

MỤC LỤC

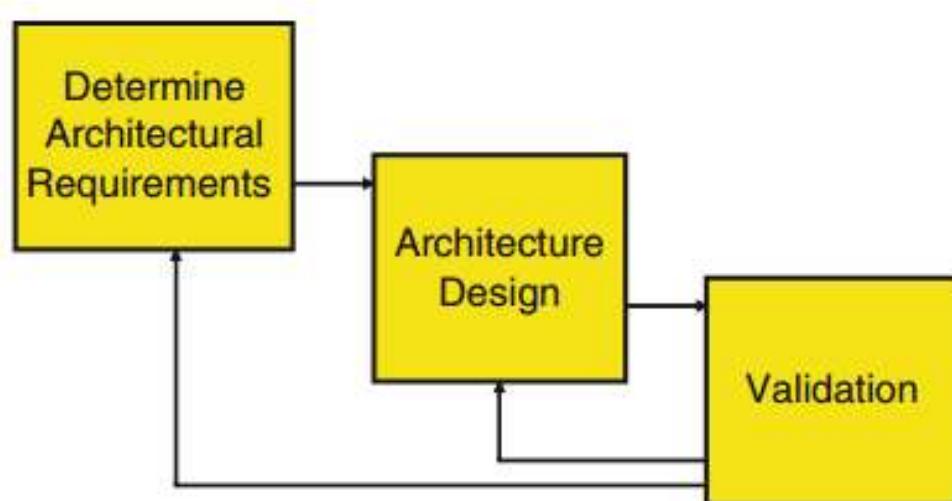
MỤC LỤC	1
QUI TRÌNH THIẾT KẾ KIẾN TRÚC	4
I. XÁC ĐỊNH CÁC YÊU CẦU KIẾN TRÚC	4
II. THIẾT KẾ KIẾN TRÚC	4
III. CHỌN LỰA KIẾN TRÚC NỀN TẢNG	5
Multitier Client Server.....	5
Messaging	5
Broker	6
Process Coordinator.....	6
Other Architectures.....	6
IV. KIỂM TRA (VALIDATION)	7
V. BÀI TẬP	7
TÀI LIỆU CHO MỘT KIẾN TRÚC PHẦN MỀM	8
I. XÁC ĐỊNH VIẾT TÀI LIỆU NHỮNG GÌ	8
II. UML 2.0	8
III. CÁC MỨC NHÌN CỦA KIẾN TRÚC	8
IV. CÁC MẪU TÀI LIỆU KIẾN TRÚC	9
MESSAGE-ORIENTED ARCHITECTURE	11
I. MICROSOFT MESSAGE QUEUE (MSMQ)	11
Cấu hình.....	11
Ứng dụng minh họa	12
II. Apache ActiveMQ.....	17
CẤU HÌNH.....	17
NGUYÊN LÝ LẬP TRÌNH	21
MÔ HÌNH POINT-TO-POINT	21
MÔ HÌNH PUBLISHER – SUBSCRIBER.....	31
SECURITY	35
MULTITIERS ARCHITECTURE	38
I. HƯỚNG DẪN LÀM VIỆC VỚI JBOSS-ECLIPSE-JAVAEE	38
CÀI ĐẶT VÀ CẤU HÌNH	38
II. STATELESS SESSION BEAN	46
EJB PROJECT.....	46
DESKTOP APPLICATION CLIENT.....	51
WEB CLIENT	56
ENTERPRISE APPLICATION PROJECT	62

BÀI TẬP	65
III. STATEFULL SESSION BEAN	66
IV. SINGLETON SESSION BEAN	67
EJB PROJECT.....	67
WEB CLIENT	69
V. MESSAGE-DRIVEN-BEAN.....	70
CẤU HÌNH.....	70
EJB PROJECT.....	71
VI. JAVA PERSISTENCE API	77
CẤU HÌNH DATASOURCE TRONG JBossEAP	77
JPA PROJECT.....	80
EJB PROJECT.....	86
CLIENT PROJECT	88
ENTERPRISE APPLICATION PROJECT	90
TRIỂN KHAI	91
BÀI TẬP	92
VII. EJB SECURITY	94
CẤU HÌNH.....	94
EJB AUTHORIZATION.....	96
EJB PROJECT.....	96
CLIENT	97
ENTERPRISE APPLICATION PROJECT	100
TRIỂN KHAI	101
HTTPS – SSL	102
BÀI TẬP	116
SERVICE-ORIENTED ARCHITECTURE	117
I. SOAP WEB SERVICE VỚI C#.....	118
TẠO WEB SERVICE	118
CLIENT	121
II. SOAP WEB SERVICE VỚI JAVA.....	124
WEBSERVICE WITH ENDPOINT	124
WEBSERVICE ON WEB SERVER.....	127
III. SOAP CLIENT WITH CODE	132
IV. SOAP WEB SERVICE TRẢ VỀ JSON FORMAT.....	135
V. WEB SERVICE SECURITY	137
SIMPLE AUTHENTICATION.....	137
VI. WSE.....	146

RESOURCE-ORIENTED ARCHITECTURE	147
I. WINDOWS COMMUNICATION FOUNDATION (WCF)	148
CẤU HÌNH IIS	148
TẠO WCF REST	148
CLIENT	155
II. JAX-RS	158
REST – JPA - JBossEAP	160
III. REST VỚI NODEJS	170
IV. SECURITY	171

QUI TRÌNH THIẾT KẾ KIẾN TRÚC (ARCHITECTURE DESIGN PROCESS)

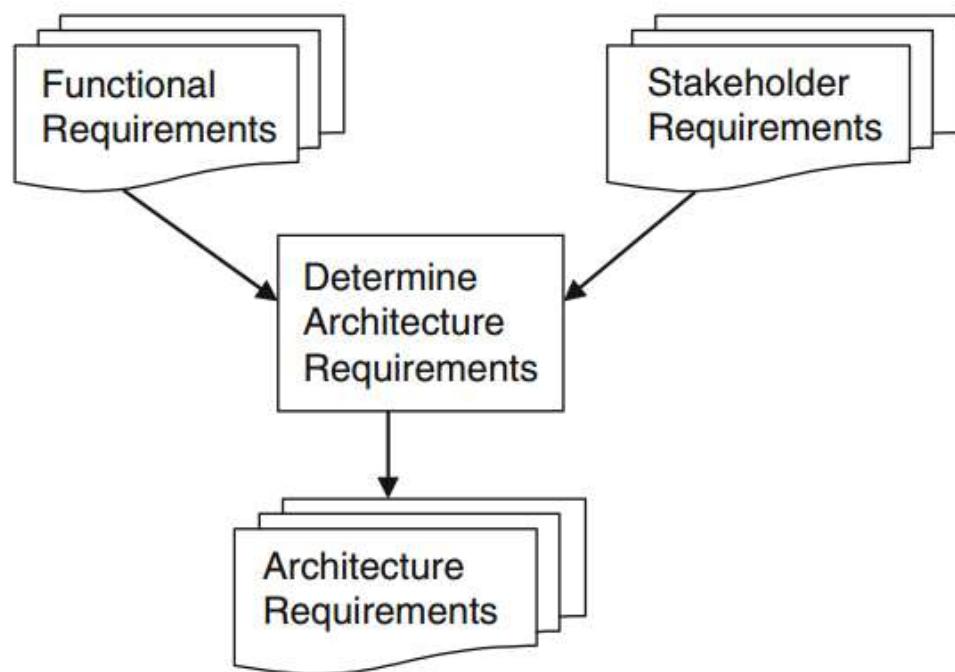
Qui trình thiết kế kiến trúc



Lưu ý về tính lặp của qui trình này.

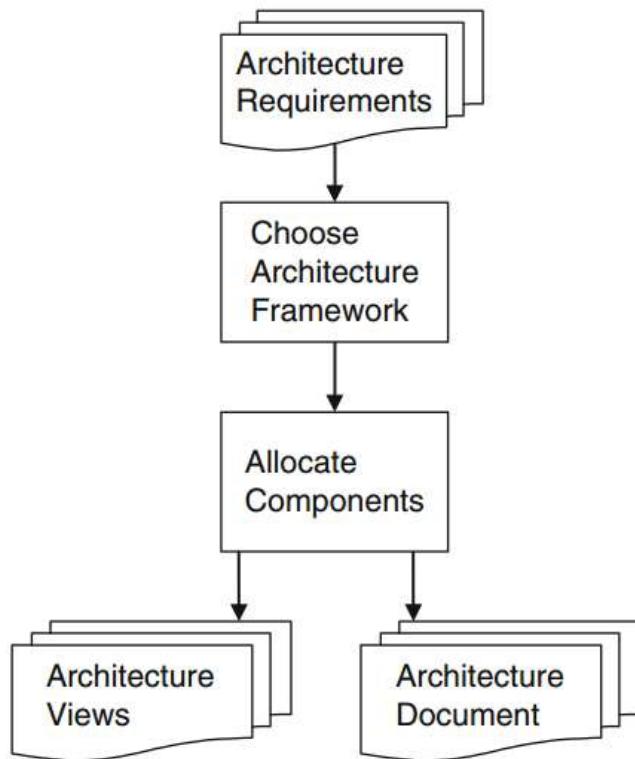
I. XÁC ĐỊNH CÁC YÊU CẦU KIẾN TRÚC

(DETERMINING ARCHITECTURE REQUIREMENTS)



II. THIẾT KẾ KIẾN TRÚC

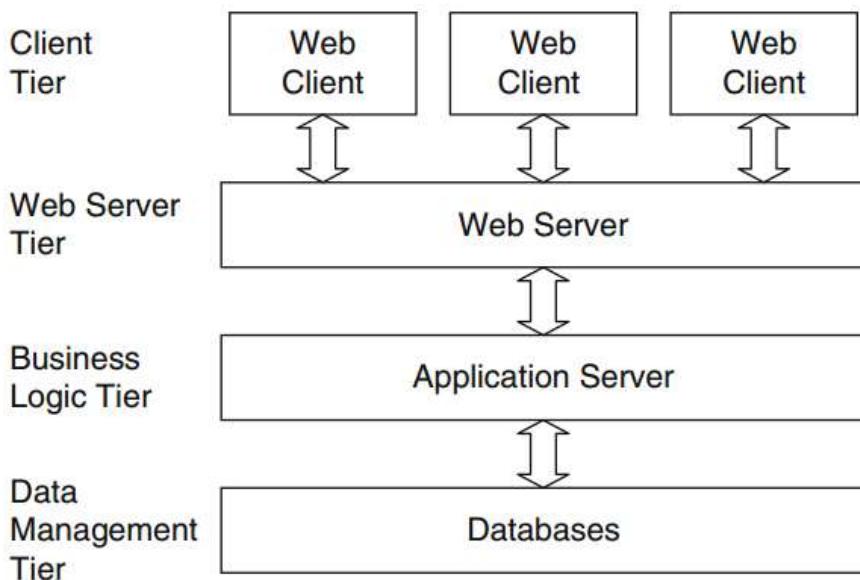
(ARCHITECTURE DESIGN)



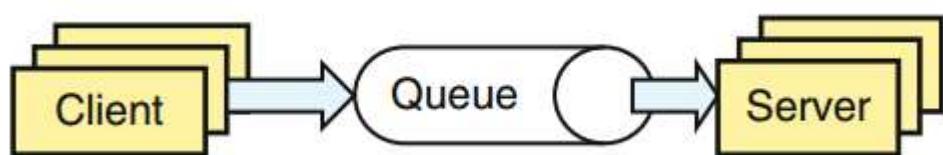
III. CHỌN LỰA KIẾN TRÚC NỀN TẢNG

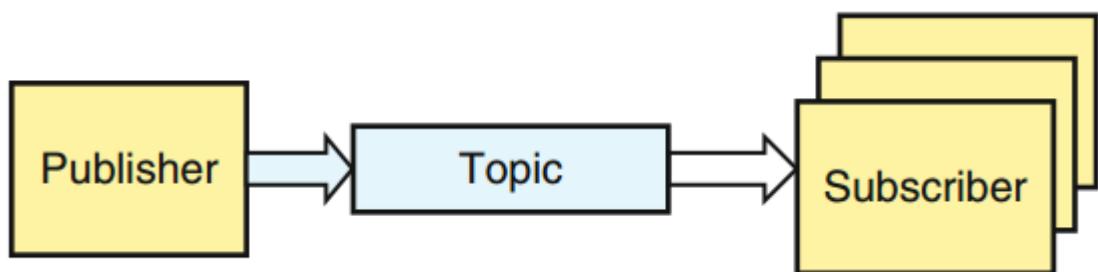
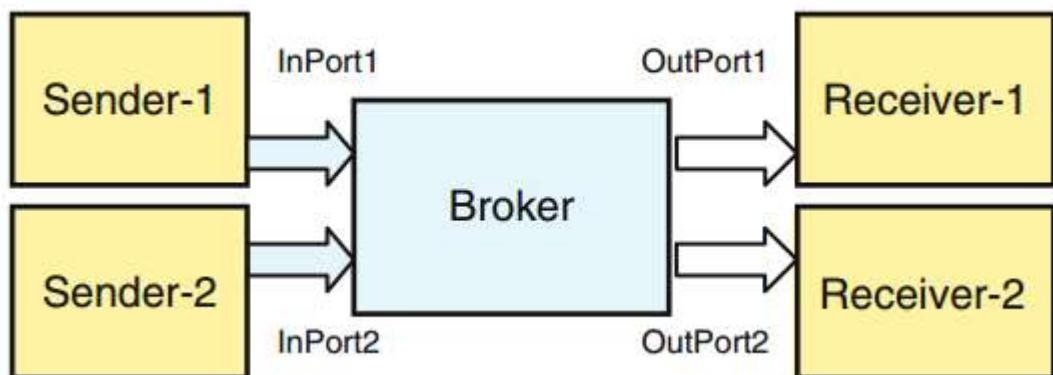
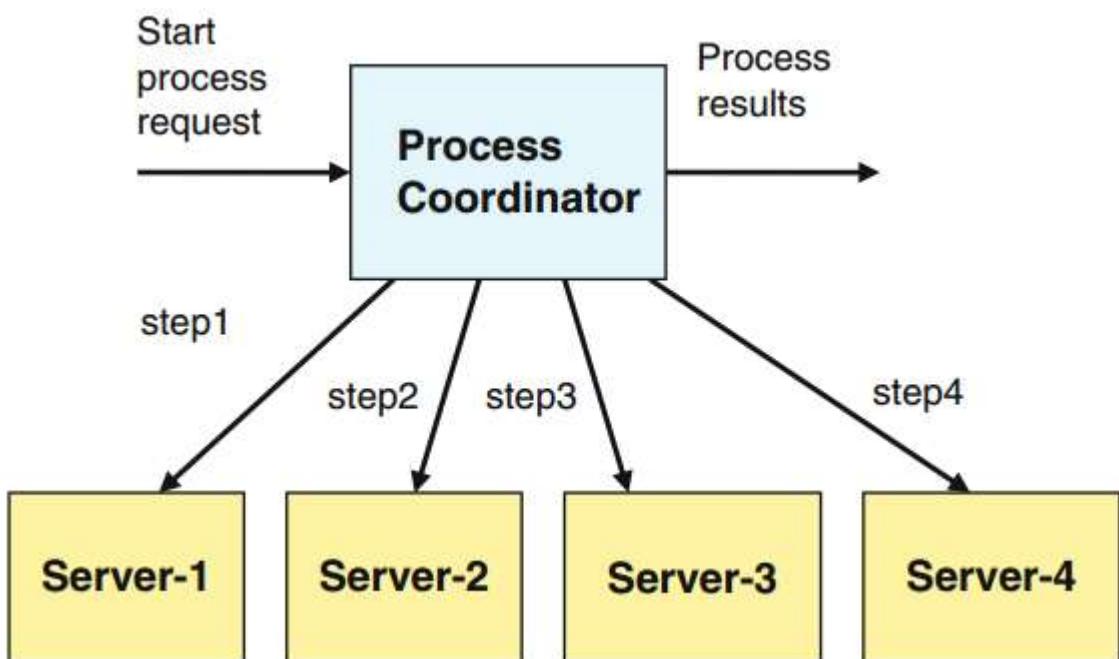
(CHOOSING THE ARCHITECTURE FRAMEWORK)

Multitier Client Server

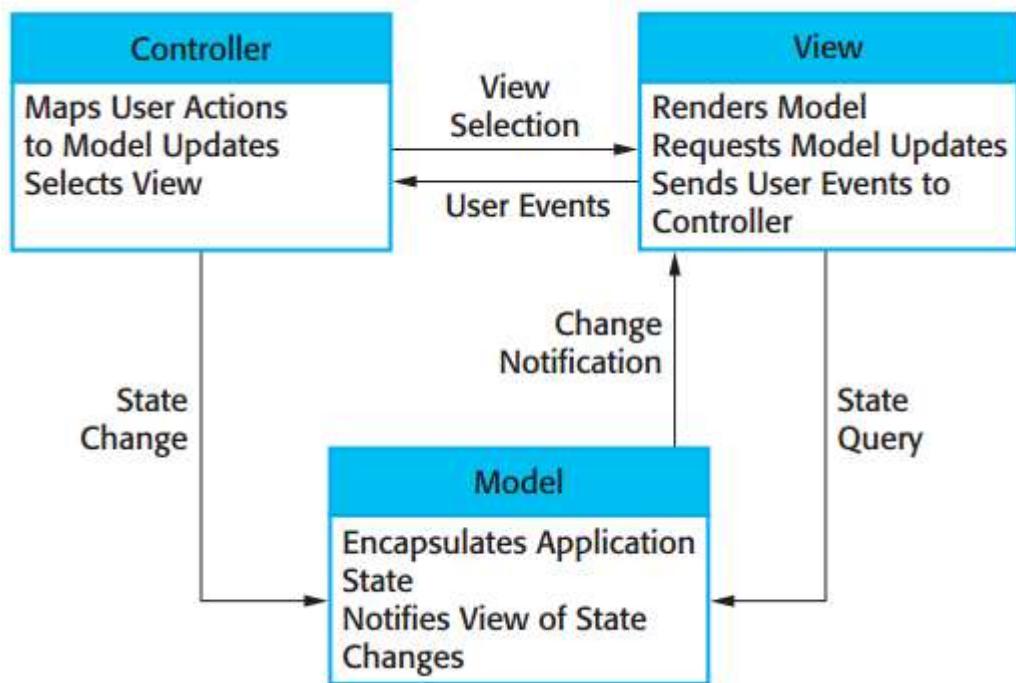


Messaging



**Broker****Process Coordinator****Other Architectures**

- Model-View-Controller



- Repository architecture
- Client-server architecture
- Pipe and filter architecture
- ...

IV. KIỂM TRA (VALIDATION)

Sử dụng các kịch bản (Scenarios)

Sử dụng các nguyên mẫu (Prototyping)

1. Proof-of-concept: Can the architecture as designed be built in a way that can satisfy the requirements?
2. Proof-of-technology: Does the technology (middleware, integrated applications, libraries, etc) selected to implement the application behave as expected?

V. BÀI TẬP

Đề xuất 1 kiến trúc, tiến hành tài liệu cho quá trình thiết kế kiến trúc

TÀI LIỆU CHO MỘT KIẾN TRÚC PHẦN MỀM

(DOCUMENTING A SOFTWARE ARCHITECTURE)

I. XÁC ĐỊNH VIẾT TÀI LIỆU NHỮNG GÌ

Điều này tùy thuộc vào độ phức tạp của hệ thống sẽ thiết kế.

Nếu là kiến trúc 2-tiers client-server, ta chỉ cần một lược đồ “**marketecture**” tổng quát mô tả các **components** của hệ thống. Thêm vào đó có thể là một mức nhìn cấu trúc của các components chính (có thể dùng kiến trúc MVC). Có thể thêm vào mô tả về lược đồ cơ sở dữ liệu.

Với một hệ thống lớn, phức tạp, có nhiều **stackholders** thì tài liệu cũng trở nên phức tạp hơn nhiều **để có thể được xem như là một công cụ giao tiếp chính**. Khi đó, các thành phần có thể sẽ tài liệu đó có thể là:

- Component interfaces
- Subsystems constraints
- Test scenarios
- Third party component purchasing decisions
- Team structure and schedule dependencies
- External services to be offered by the application

Nên nhớ: việc viết và duy trì tài liệu sẽ tốn rất nhiều công sức và tiền bạc. Vì thế cho nên phải cân nhắc những gì sẽ được viết sao cho tốt nhất

II. UML 2.0

- Use-case diagram
- Activity diagram
- Sequence diagram
- Component diagram
- Deployment diagram
- Class diagram
- Và các diagrams khác

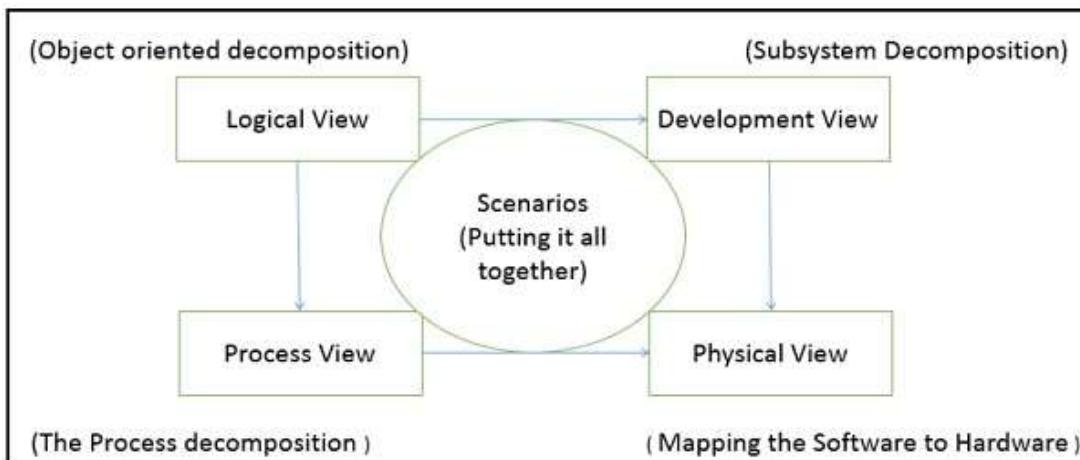
Cũng cần phải nói thêm rằng: UML chỉ hữu ích trong qui trình phát triển hướng mô hình (model-driven development) còn trong xu hướng hiện nay phát triển theo hướng linh hoạt (agile) thì tác dụng không nhiều.

III. CÁC MỨC NHÌN CỦA KIẾN TRÚC

Krutch (1995) 4+1 model views:

1. **A logical view**, which shows the key abstractions in the system as objects or object classes. It should be possible to relate the system requirements to entities in this logical view.
2. **A process view**, which shows how, at run-time, the system is composed of interacting processes. This view is useful for making judgments about nonfunctional system characteristics such as performance and availability.
3. **A development view**, which shows how the software is decomposed for development, that is, it shows the breakdown of the software into components that are implemented by a single developer or development team. This view is useful for software managers and programmers.

4. A **physical view**, which shows the system hardware and how software components are distributed across the processors in the system. This view is useful for systems engineers planning a system deployment
5. Related using use cases or scenarios (+1)



IV. CÁC MẪU TÀI LIỆU KIẾN TRÚC

(Architecture Documentation Template)

- ◀ 1. Introduction
 - 1.1 Purpose
 - 1.2 Scope
- 1.3 Team Members and Roles
- 1.4 Definitions, Acronyms, and Abbreviations
- 1.5 References
- 2. Architectural Goals and Constraints
- ◀ 3. High-level Architectural Overview
 - 3.1 Dependencies on other Systems
 - 3.2 Dependencies on Data stores
- ◀ 4. Use-Case View
 - 4.1 Overview
 - 4.2 Use-Case Realizations
- ◀ 5. Logical View
 - 5.1 Overview
 - 5.2 Analysis Model
 - 5.3 Significant Packages or Subsystems or Layers or Tiers
- ◀ 6. Process View
 - 6.1 Overview
 - 6.2 Interprocess Communication Mechanisms
- ◀ 7. Deployment View
 - 7.1 Overview
 - 7.2 Deployment Policies
- ◀ 8. Data View
 - 8.1 Overview
 - 8.2 Data Access Mechanisms
- 9. Size and Performance
- 10. Quality
- 11. Common Patterns and Mechanisms

--Tham khảo thêm trên internet--

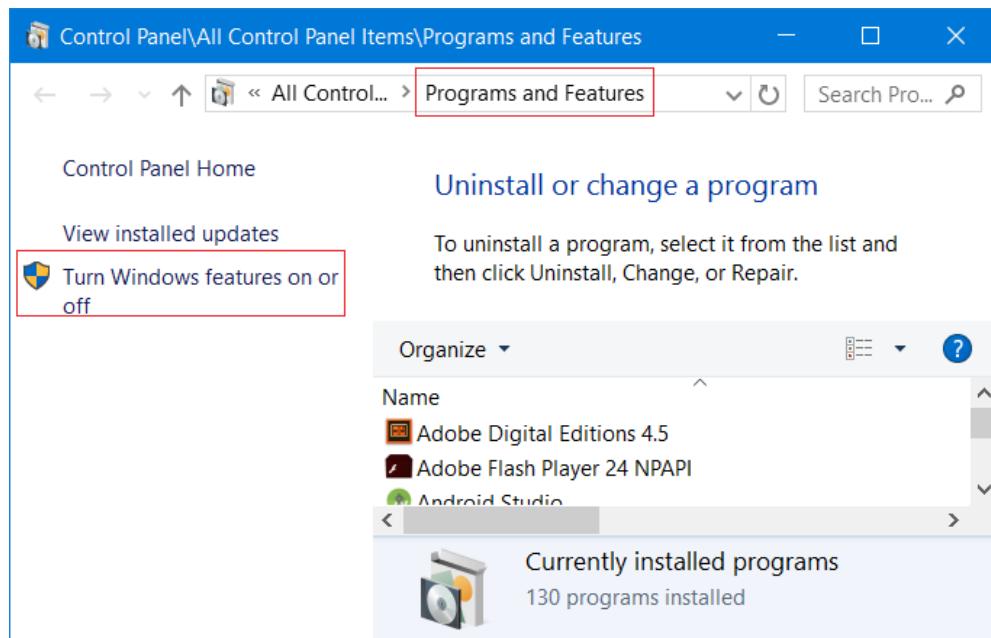
MESSAGE-ORIENTED ARCHITECTURE

I. MICROSOFT MESSAGE QUEUE (MSMQ)

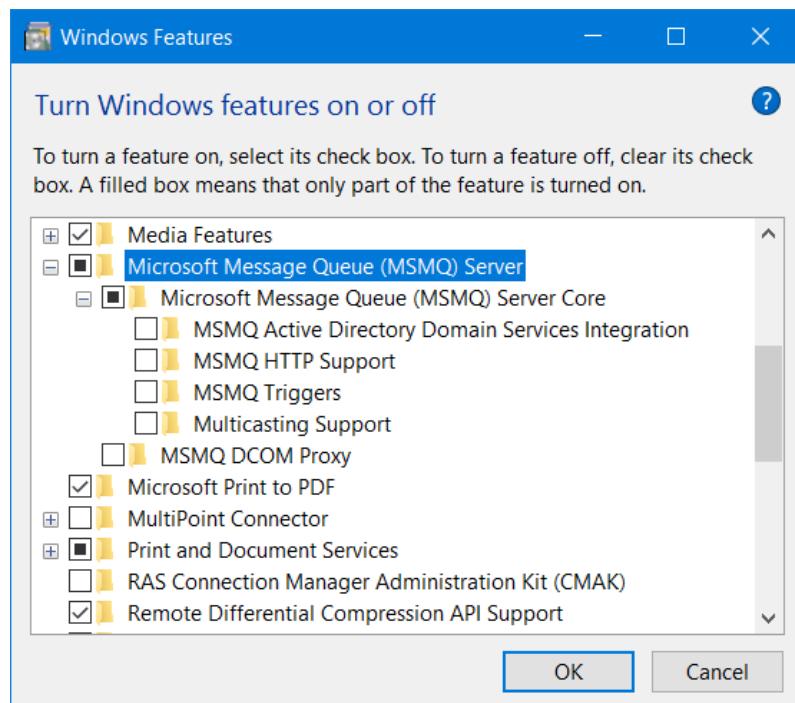
Cấu hình

Hướng dẫn cấu hình và làm việc với MSMQ trong Windows

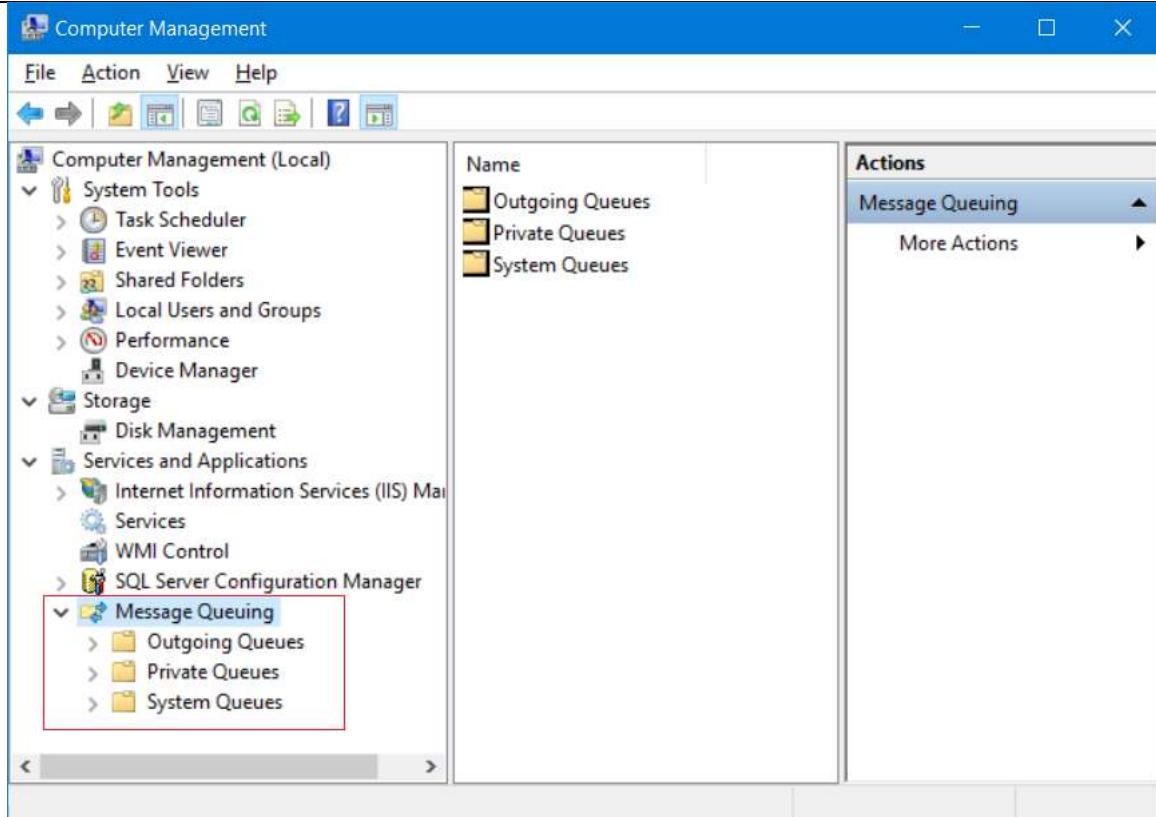
1. Mở Control Panel, chọn “Programs and Features” rồi chọn “Turn Windows features on or off”



Chọn như hình



2. Mở Computer Management bằng cách nhấn chuột phải lên My Computer rồi chọn Manage. Sau đó mở rộng chọn như hình. Quan sát kết quả.

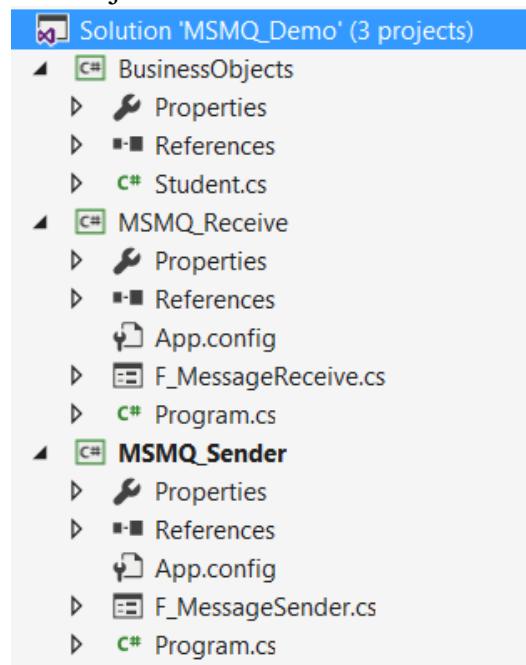


Ứng dụng minh họa

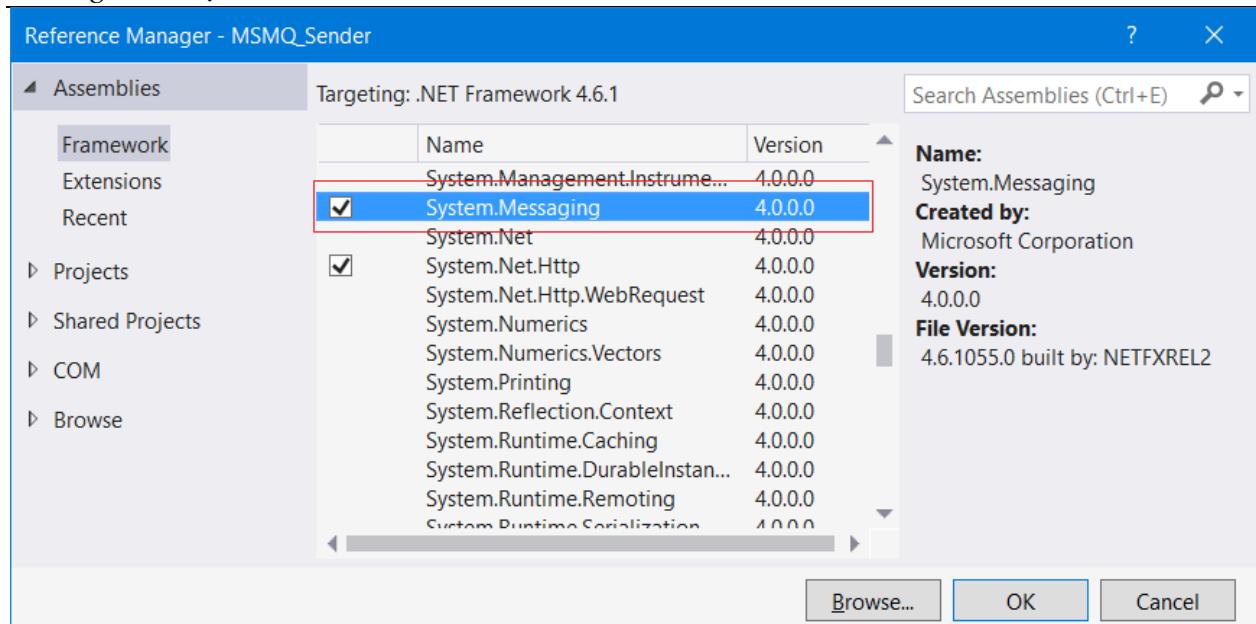
Trong ví dụ này chúng ta làm một ứng dụng về gửi/nhận message dạng text và dạng object

Tạo 3 Project C# mới:

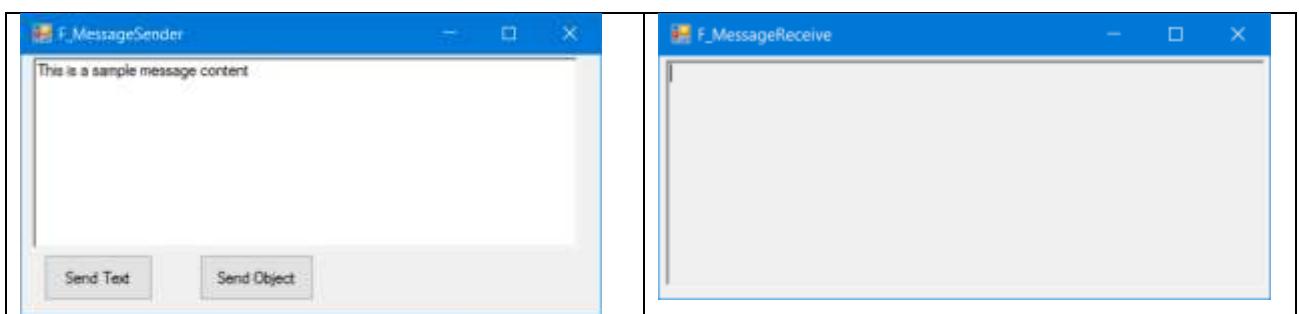
- Một cho việc gửi message
- Một cho việc nhận message.
- Một để chứa business objects



Thêm Reference đến System.Messaging cho 2 projects gửi và nhận như hình



Giao diện:



Code cho đối tượng dùng để gửi (business object)

```
using System;
namespace BusinessObjects
{
    [Serializable] //chú ý đối tượng phải được Serialize
    public class Student
    {
        public long StudentId { get; set; }
        public string FullName { get; set; }
        public DateTime DOB { get; set; }
        public Student():this(0,"no-name",new DateTime())
        {
        }
        public Student(long id,string fname, DateTime dob)
        {
            StudentId = id;FullName = fname;DOB = dob;
        }
        public override string ToString()
        {
            return FullName+"\t"+DOB;
        }
    }
}
```

Code cho form send message

```
using BusinessObjects;
using System;
using System.Messaging;
using System.Windows.Forms;
```

```

namespace MSMQ_Sender
{
    public partial class F_MessageSender : Form
    {
        MessageQueue queue = null;
        public F_MessageSender()
        {
            InitializeComponent();
            init();
        }

        private void init()
        {
            string path = @".\private$\phongkehoach";
            //string path = @"hbmn1\private$\phongkehoach";
            if (MessageQueue.Exists(path))
            {
                queue = new MessageQueue(path, QueueAccessMode.Send);
            }
            else
                queue = MessageQueue.Create(path, true);
            queue.Label = "queue cho phong ke hoach";
        }
        private void sendButton_Click(object sender, EventArgs e)
        {
            string message = richTextBox1.Text;
            MessageQueueTransaction transaction = new MessageQueueTransaction();
            transaction.Begin();
            queue.Send(message, transaction);
            transaction.Commit();
        }
        private void SendObjectButton_Click(object sender, EventArgs e)
        {
            Student st=new Student(1001L,"Nguyễn văn Tèo",new DateTime(1999,10,15));
            MessageQueueTransaction transaction = new MessageQueueTransaction();
            transaction.Begin();
            queue.Send(st, transaction);
            transaction.Commit();
        }
    }
}

```

Code cho form receive message

```

using BusinessObjects;
using System.Messaging;
using System.Windows.Forms;
namespace MSMQ_Receive
{
    public partial class F_MessageReceive : Form
    {
        private MessageQueue queue;
        public F_MessageReceive()
        {
            InitializeComponent();
            init_queue();
        }
        void init_queue()
        {
            string path = @".\private$\phongkehoach";
            queue = new MessageQueue(path);
            queue.BeginReceive();
            queue.ReceiveCompleted += Queue_ReceiveCompleted;
        }
    }
}

```

```
private void Queue_ReceiveCompleted(object sender, ReceiveCompletedEventArgs e)
{
    var msg = e.Message;
    int type = msg.BodyType;
    XmlMessageFormatter fmt = new XmlMessageFormatter(
        new System.Type[]{ typeof(string),typeof(Student) }
    );
    msg.Formatter = fmt;
    var result=msg.Body;
    var t=result.GetType();
    if (t.Equals(typeof(Student)))
    {
        SetText(t + ":" + result);
    }
    else
        SetText(""+result);
    queue.BeginReceive(); //loop back
}

delegate void SetTextCallback(string text);
private void SetText(string text)
{
    // InvokeRequired required compares the thread ID of the
    // calling thread to the thread ID of the creating thread.
    // If these threads are different, it returns true.
    if (this.MessagesRichTextBox.InvokeRequired)
    {
        SetTextCallback callback = new SetTextCallback(SetText);
        this.Invoke(callback, new object[] { text });
    }
    else
    {
        this.MessagesRichTextBox.AppendText(text + "\n");
    }
}
```

Thực thi đồng thời hai chương trình rồi nhập nội dung cần gửi sau đó nhấn nút gửi. Quan sát kết quả.

TRUY XUẤT QUA MẠNG

Cấu hình cho 2 máy có thẻ truy xuất qua mạng (LAN/internet)

Chú ý đường dẫn

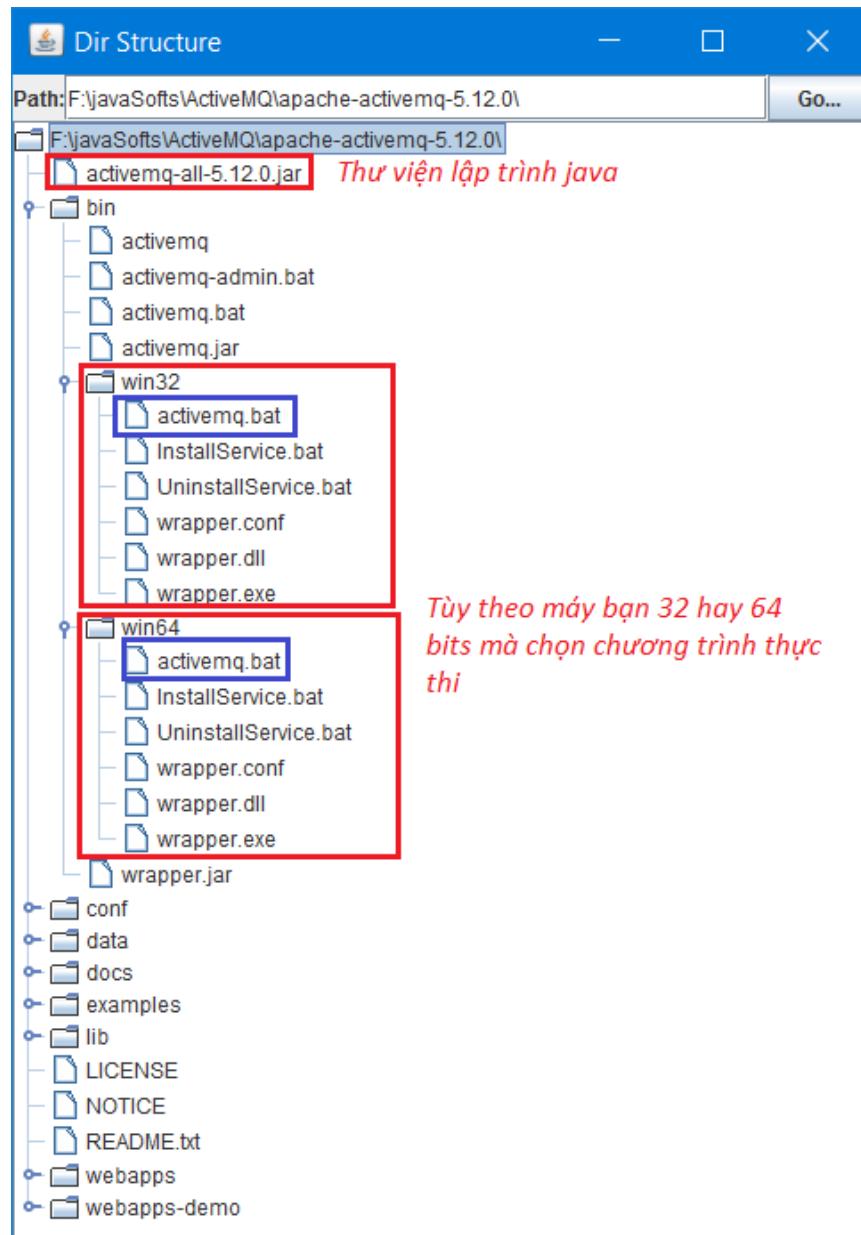
```
string path = @"FormatName:Direct:OS:serverName\private$\queue_name";
hoặc
string path = @"FormatName:Direct:TCP:serverIPaddress\private$\queue_name";
```

Nếu truy xuất server từ xa, ta không thể kiểm tra queue có tồn tại hay không cũng như tạo queue mới, nên code phải modify lại

```
private void init()
{
    string path = @" FormatName:Direct:OS:serverName\private$\queue_name";
    queue = new MessageQueue(path, QueueAccessMode.Send);
}
```

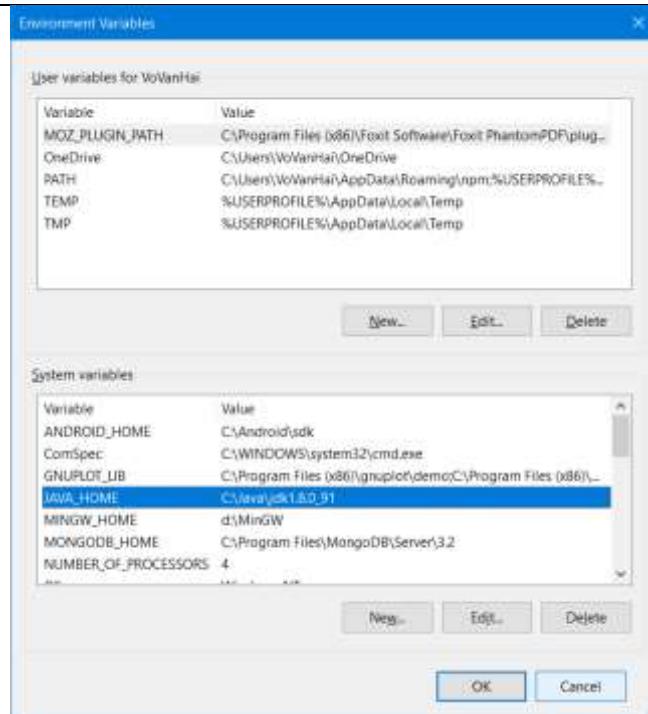

II. Apache ActiveMQ

Download tại <http://activemq.apache.org/> sau đó tiến hành giải nén ra một thư mục nào đó (giả sử F:\javaSofts\ActiveMQ)



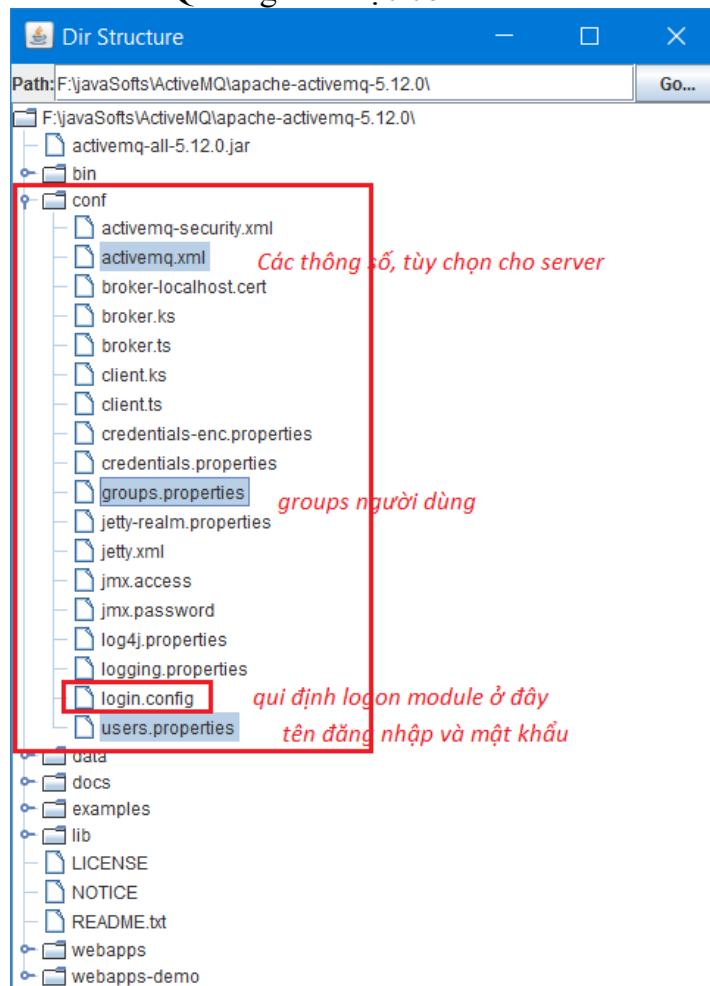
CẤU HÌNH

1. Cần cấu hình biến môi trường JAVA_HOME là thư mục nơi cài đặt JDK. Ví dụ như sau



2. Cấu hình cho server

Các file cấu hình của ActiveMQ trong thư mục conf



Cấu hình login module (bắt buộc login khi kết nối)

File **login.config**

```
activemq {
    org.apache.activemq.jaas.PropertiesLoginModule required
```

```
org.apache.activemq.jaas.properties.user="users.properties"
org.apache.activemq.jaas.properties.group="groups.properties"
reload=true;
};
```

Cấu hình groups, file **groups.properties**

```
#group_name=users_list
admins=admin,teo
users=ty
dhcn=men
guests=guest
```

Cấu hình người dùng, file **users.properties**

```
#username=password
admin=admin
teo=123
ty=456
men=123
```

Đọc thêm phần security ở đây: <http://activemq.apache.org/security.html>

3. Thực thi

Chạy file activemq.bat trong thư mục Win32 hoặc Win64 tùy theo máy 32 hay 64 bits.

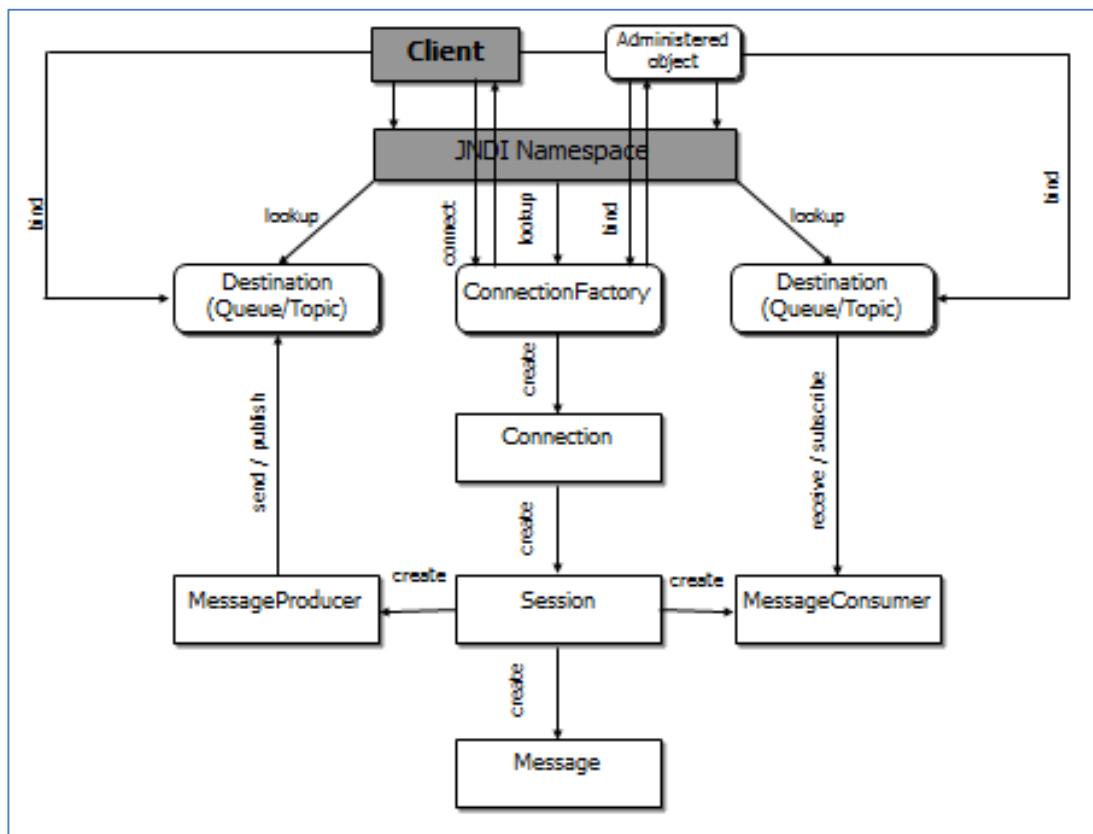
```
jvm 1 | INFO | Apache ActiveMQ 5.12.0 (localhost, ID:HBMLN-7069-1486990366963-0:1) is starting
jvm 1 | INFO | Listening for connections at: tcp://HBMLN:61616?maximumConnections=1000&wireFormat.maxFrameSize=1048576
jvm 1 | INFO | Connector openwire started
jvm 1 | INFO | Listening for connections at: amqp://HBMLN:5672?maximumConnections=1000&wireFormat.maxFrameSize=1048576
jvm 1 | INFO | Connector amqp started
jvm 1 | INFO | Listening for connections at: stomp://HBMLN:61613?maximumConnections=1000&wireFormat.maxFrameSize=1048576
jvm 1 | INFO | Connector stomp started
jvm 1 | INFO | Listening for connections at: mqtt://HBMLN:1883?maximumConnections=1000&wireFormat.maxFrameSize=1048576
jvm 1 | INFO | Connector mqtt started
jvm 1 | {}
jvm 1 | INFO | Listening for connections at ws://HBMLN:61614?maximumConnections=1000&wireFormat.maxFrameSize=1048576
jvm 1 | INFO | Connector ws started
jvm 1 | INFO | Apache ActiveMQ 5.12.0 (localhost, ID:HBMLN-7069-1486990366963-0:1) started
jvm 1 | INFO | For help or more information please see: http://activemq.apache.org
jvm 1 | WARN | Store limit is 102400 mb (current store usage is 4 mb). The data directory: F:\javaSofts\ActiveMQ\apache-activemq-5.12.0\bin\win64\..\..\data\kahadb only has 68468 mb of usable space - resetting to maximum available disk space: 68473 mb
jvm 1 | INFO | ActiveMQ WebConsole available at http://0.0.0.0:8161/
jvm 1 | INFO | ActiveMQ Jolokia REST API available at http://0.0.0.0:8161/api/jolokia/
jvm 1 | INFO | Initializing Spring FrameworkServlet 'dispatcher'
jvm 1 | INFO | jolokia-agent: No access restrictor found at classpath:/jolokia-access.xml, access to all MBeans is allowed
```

(lưu ý: không đóng cửa sổ này)

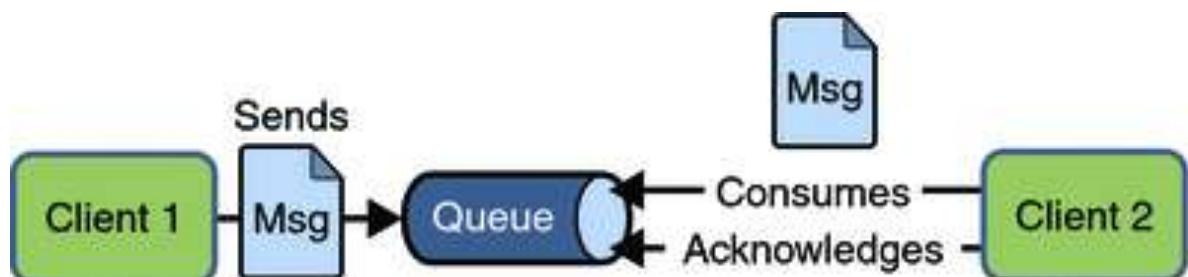
Mở cửa sổ browser, gõ địa chỉ: <http://localhost:8161/admin/>. Nếu có yêu cầu đăng nhập gõ username và password đều là admin

The screenshot shows a web browser window displaying the Apache ActiveMQ Admin Console at localhost:8161/admin/. The title bar says "localhost : ActiveMQ Cons...". The main content area features the ActiveMQ logo and the Apache Software Foundation logo. A navigation menu at the top includes Home, Queues, Topics, Subscribers, Connections, Network, Scheduled, Send, and Support. On the right side, there is a sidebar with sections for Queue Views (Graph, XML), Topic Views (XML), Subscribers Views (XML), and Useful Links (Documentation, FAQ, Downloads). The central content area displays a "Welcome!" message, information about the broker (Name: localhost, Version: 5.12.0, ID: HBMNL-7069-1486990366963-0:1), and links to the Apache ActiveMQ Site.

4. Đọc thêm tài liệu cấu hình tại <http://activemq.apache.org/getting-started.html>

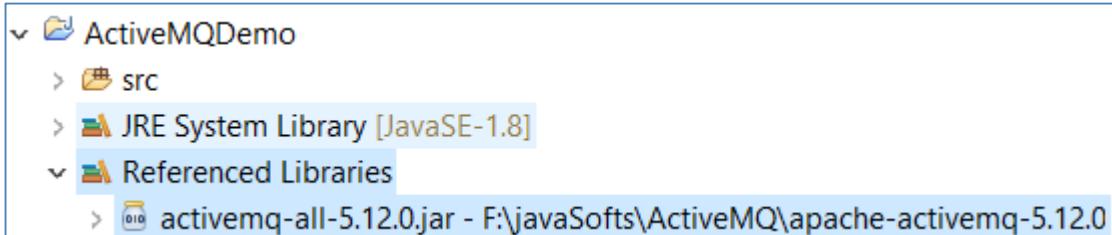


MÔ HÌNH POINT-TO-POINT



CODE MINH HỌA CHO GỬI VÀ NHẬN BẰNG JAVA

- Khởi động eclipse, chọn perspective là Java. Tạo project mới
- Thêm tham chiếu đến thư viện lập trình java của ActiveMQ



- Code helper

```

package helper;

import java.io.StringReader;
import java.io.StringWriter;
import javax.xml.bind.JAXBContext;
import javax.xml.bind.Marshaller;
import javax.xml.bind.Unmarshaller;
  
```

```

public class XMLConvert<T> {
    private T type;

    public XMLConvert(T type) {
        this.type = type;
    }

    @SuppressWarnings("all")
    public T xml2Object(String xml) throws Exception{
        T sv=null;
        JAXBContext ctx= JAXBContext.newInstance(type.getClass());
        Unmarshaller ms = ctx.createUnmarshaller();
        sv=(T) ms.unmarshal(new StringReader(xml));
        return sv;
    }

    public String object2XML(T obj) throws Exception{
        JAXBContext ctx= JAXBContext.newInstance(type.getClass());
        Marshaller ms=ctx.createMarshaller();
        StringWriter sw = new StringWriter();
        ms.marshal(obj, sw);
        return sw.toString();
    }
}

```

Sample object

```

package data;
import java.io.Serializable;
import java.util.Date;
import javax.xml.bind.annotation.XmlRootElement;
import javax.xml.bind.annotation.XmlType;

@XmlRootElement
@XmlType(propOrder={"mssv","hoten","ngaysinh"})
public class Person implements Serializable{
    private long mssv;
    private String hoten;
    private Date ngaysinh;
    public Person(long mssv, String hoten, Date ngaysinh) {
        this.mssv = mssv;
        this.hoten = hoten;
        this.ngaysinh = ngaysinh;
    }
    public Person() {
    }
    public long getMssv() {
        return mssv;
    }
    public void setMssv(long mssv) {
        this.mssv = mssv;
    }
    public String getHoten() {
        return hoten;
    }
    public void setHoten(String hoten) {
        this.hoten = hoten;
    }
    public Date getNgaysinh() {
        return ngaysinh;
    }
    public void setNgaysinh(Date ngaysinh) {
        this.ngaysinh = ngaysinh;
    }
}

```

```

    }
    @Override
    public String toString() {
        return mssv+"\t"+hoten+"\t"+ngaysinh;
    }
}

```

4. Code Sender

```

import java.util.Date;
import java.util.Properties;
import javax.jms.Connection;
import javax.jms.ConnectionFactory;
import javax.jms.Destination;
import javax.jms.Message;
import javax.jms.MessageProducer;
import javax.jms.Session;
import javax.naming.Context;
import javax.naming.InitialContext;
import org.apache.log4j.BasicConfigurator;
import data.Person; import helper.XMLConvert;

public class QueueSender {
    public static void main(String[] args) throws Exception{
        //config environment for JMS
        BasicConfigurator.configure();
        //config environment for JNDI
        Properties settings=new Properties();
        settings.setProperty(Context.INITIAL_CONTEXT_FACTORY,
                "org.apache.activemq.jndi.ActiveMQInitialContextFactory");
        settings.setProperty(Context.PROVIDER_URL, "tcp://localhost:61616");
        //create context
        Context ctx=new InitialContext(settings);
        //lookup JMS connection factory
        ConnectionFactory factory=
                (ConnectionFactory)ctx.lookup("ConnectionFactory");
        //lookup destination. (If not exist-->ActiveMQ create once)
        Destination destination=
                (Destination) ctx.lookup("dynamicQueues/thanthidet");
        //get connection using credential
        Connection con=factory.createConnection("admin","admin");
        //connect to MOM
        con.start();
        //create session
        Session session=con.createSession(
                /*transaction*/false,
                /*ACK*/Session.AUTO_ACKNOWLEDGE
        );
        //create producer
        MessageProducer producer = session.createProducer(destination);
        //create text message
        Message msg=session.createTextMessage("hello mesage from ActiveMQ");
        producer.send(msg);

        Person p=new Person(1001, "Thân Thị Đẹt", new Date());
        String xml=new XMLConvert<Person>(p).object2XML(p);

        msg=session.createTextMessage(xml);
        producer.send(msg);
        //shutdown connection
        session.close();con.close();
        System.out.println("Finished...");
    }
}

```

```

    }
}
```

5. Code Receiver

```

package demo01.ex01;

import java.util.Properties;
import javax.jms.Connection;
import javax.jms.ConnectionFactory;
import javax.jms.Destination;
import javax.jms.Message;
import javax.jms.MessageConsumer;
import javax.jms.MessageListener;
import javax.jms.ObjectMessage;
import javax.jms.Session;
import javax.jms.TextMessage;
import javax.naming.Context;
import javax.naming.InitialContext;
import org.apache.log4j.BasicConfigurator;

public class QueueReceiver {
    public static void main(String[] args) throws Exception{
        //thiết lập môi trường cho JMS
        BasicConfigurator.configure();
        //thiết lập môi trường cho JNDI
        Properties settings=new Properties();
        settings.setProperty(Context.INITIAL_CONTEXT_FACTORY,
                            "org.apache.activemq.jndi.ActiveMQInitialContextFactory");
        settings.setProperty(Context.PROVIDER_URL, "tcp://localhost:61616");
        //tạo context
        Context ctx=new InitialContext(settings);
        //lookup JMS connection factory
        Object obj=ctx.lookup("ConnectionFactory");
        ConnectionFactory factory=(ConnectionFactory)obj;
        //lookup destination
        Destination destination
        =(Destination) ctx.lookup("dynamicQueues/thanthidet");
        //tạo connection
        Connection con=factory.createConnection("admin", "admin");
        //nối đến MOM
        con.start();
        //tạo session
        Session session=con.createSession(
            /*transaction*/false,
            /*ACK*/Session.CLIENT_ACKNOWLEDGE
        );
        //tạo consumer

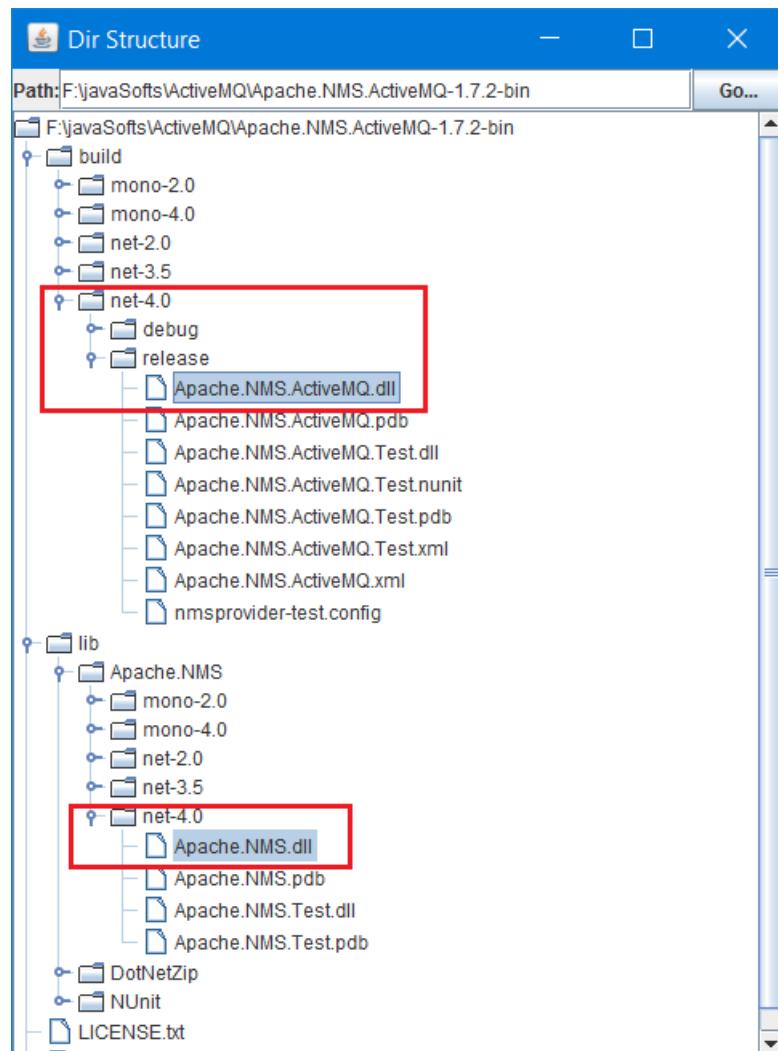
        MessageConsumer receiver = session.createConsumer(destination);
        //blocked-method for receiving message - sync
        //receiver.receive();
        //Cho receiver lắng nghe trên queue, chúng có message thì notify - async
        System.out.println("Tý was listened on queue... ");
        receiver.setMessageListener(new MessageListener() {
            @Override
            //có message đến queue, phương thức này được thực thi
            public void onMessage(Message msg) {//msg là message nhận được
                try {
                    if(msg instanceof TextMessage){
                        TextMessage tm=(TextMessage)msg;
                        String txt=tm.getText();
                        System.out.println("Nhận được "+txt);
                }
            }
        });
    }
}
```

```
msg.acknowledge(); //gửi tín hiệu ack
}
else if(msg instanceof ObjectMessage){
    ObjectMessage om=(ObjectMessage)msg;
    System.out.println(om);
}
//others message type....
} catch (Exception e) {
    e.printStackTrace();
}
}
});
```

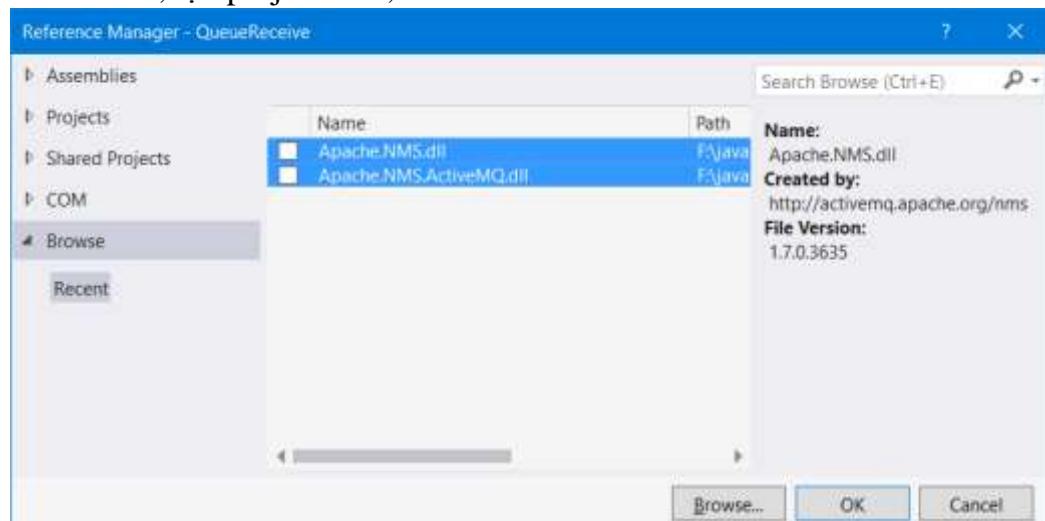
CODE MINH HỌA CHO GỬI VÀ NHẬN BẰNG C#

1. Download thư viện dùng cho .Net tại <https://cwiki.apache.org/confluence/display/NMS/ActiveMQ+Downloads>

Giải nén ta có cấu trúc thư mục như hình sau. Chú ý các chỗ đánh dấu là 2 thư viện chính ta cần làm việc.



2. Mở Visual Studio, tạo project mới, thêm tham chiếu như hình



3. Code helper

```
using System;
```

```

namespace ActiveMQSender
{
    [Serializable]
    public class Person
    {
        public long MSSV { get; set; }
        public string Hoten { get; set; }
        public DateTime Ngaysinh { get; set; }

        public Person() { }

        public Person(long mssv, string hoten, DateTime ngaysinh)
        {
            this.MSSV = mssv; Hoten = hoten; Ngaysinh = ngaysinh;
        }
    }
}

using System.Xml.Serialization;
using System.IO;
namespace ActiveMQSender
{
    public class XMLObjectConverter<T>
    {
        public string object2XML(T p)
        {
            string xml = "";
            XmlSerializer ser = new XmlSerializer(typeof(T));
            using (MemoryStream ms = new MemoryStream())
            {
                ser.Serialize(ms, p);
                ms.Position = 0;
                xml = new StreamReader(ms).ReadToEnd();
            }
            return xml;
        }
    }
}

```

4. Code cho sender

```

using System;
using System.IO;
using Apache.NMS;
using Apache.NMS.ActiveMQ;
using Apache.NMS.ActiveMQ.Commands;

namespace ActiveMQSender
{
    class Program
    {
        static void Main(string[] args)
        {

            Console.WriteLine("sending message. Enter to exit.");
            //tạo connection factory
            IConnectionFactory factory = new ConnectionFactory("tcp://localhost:61616");
            //tạo connection
            IConnection con = factory.CreateConnection("admin", "admin");
            con.Start(); //nối tới MOM
            //tạo session
            ISession session = con.CreateSession(AcknowledgementMode.AutoAcknowledge);
            //tạo producer
            ActiveMQQueue destination = new ActiveMQQueue("thanhidet");
            IMessageProducer producer = session.CreateProducer(destination);
        }
    }
}

```

```

    //send message
    //biến đổi tượng thành XML document String
    Person p = new Person(1001, "Truong Van Coi", new DateTime());
    //string xml = genXML(p).ToLower();
    string xml = new XMLObjectConverter<Person>().object2XML(p);

    Console.WriteLine(xml.ToLower());
    IMessage msg = new ActiveMQTextMessage("Hola mundo");
    producer.Send(msg);

    //shutdown
    session.Close();
    con.Close();
    Console.ReadKey();
}
}
}

```

5. Code cho Receiver

```

using System;
using Apache.NMS;
using Apache.NMS.ActiveMQ;
using Apache.NMS.ActiveMQ.Commands;

namespace ActiveMQDemo
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("receiving message. Enter to exit.");
            //tạo connection factory
            IConnectionFactory factory = new ConnectionFactory("tcp://localhost:61616");
            //tạo connection
            IConnection con = factory.CreateConnection("admin", "admin");
            con.Start(); //nối tới MOM
            //tạo session
            ISession session = con.CreateSession(AcknowledgementMode.AutoAcknowledge);
            //tạo consumer
            ActiveMQQueue destination = new ActiveMQQueue("thanhidet");
            IMessageConsumer consumer = session.CreateConsumer(destination);
            //nhận mesage - lắng nghe
            consumer.Listener += Consumer_Listener;
            Console.ReadKey();
        }

        private static void Consumer_Listener(IMessage message)
        {
            if(message is ActiveMQTextMessage)
            {
                ActiveMQTextMessage msg = message as ActiveMQTextMessage;
                Console.WriteLine("receive:" + msg.Text);
            }
        }
    }
}

```

6. Thực thi xem kết quả

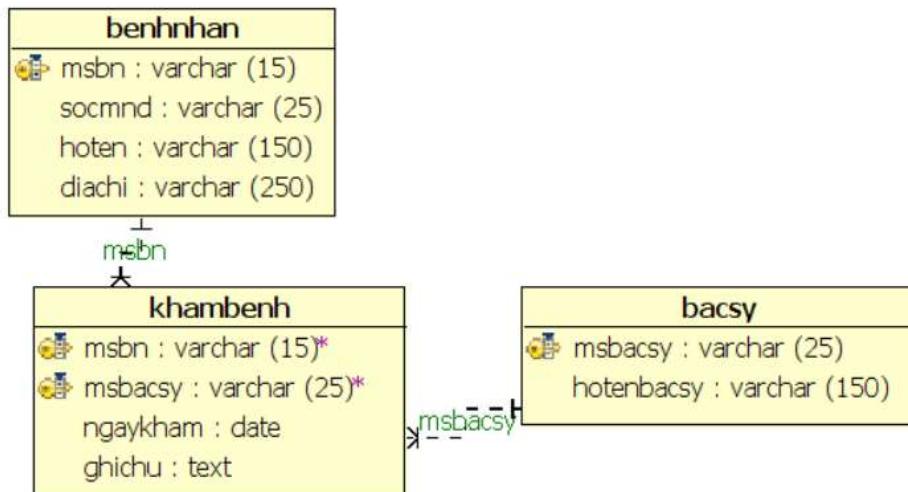
THỰC THI CHÉO

1. Chạy Sender của Java, Receiver của C#
2. Chạy Sender của C#, Receiver của Java

Quan sát kết quả

BÀI TẬP

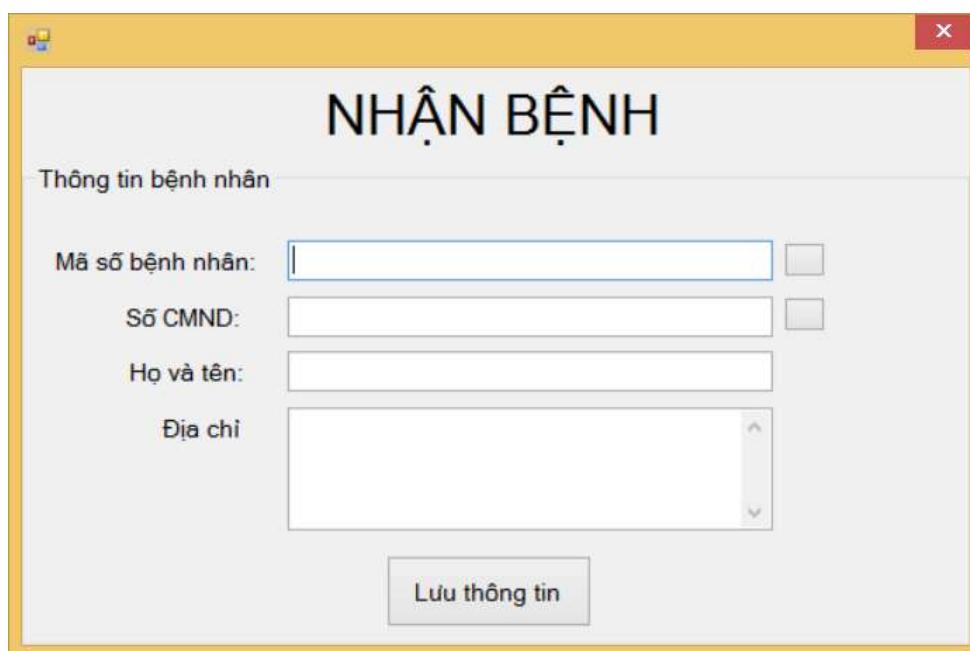
1. Cho cơ sở dữ liệu quản lý thông tin bệnh nhân được cho như sau



Yêu cầu:

Chương trình được viết cho 2 loại nhân sự của phòng khám đặt tả như sau

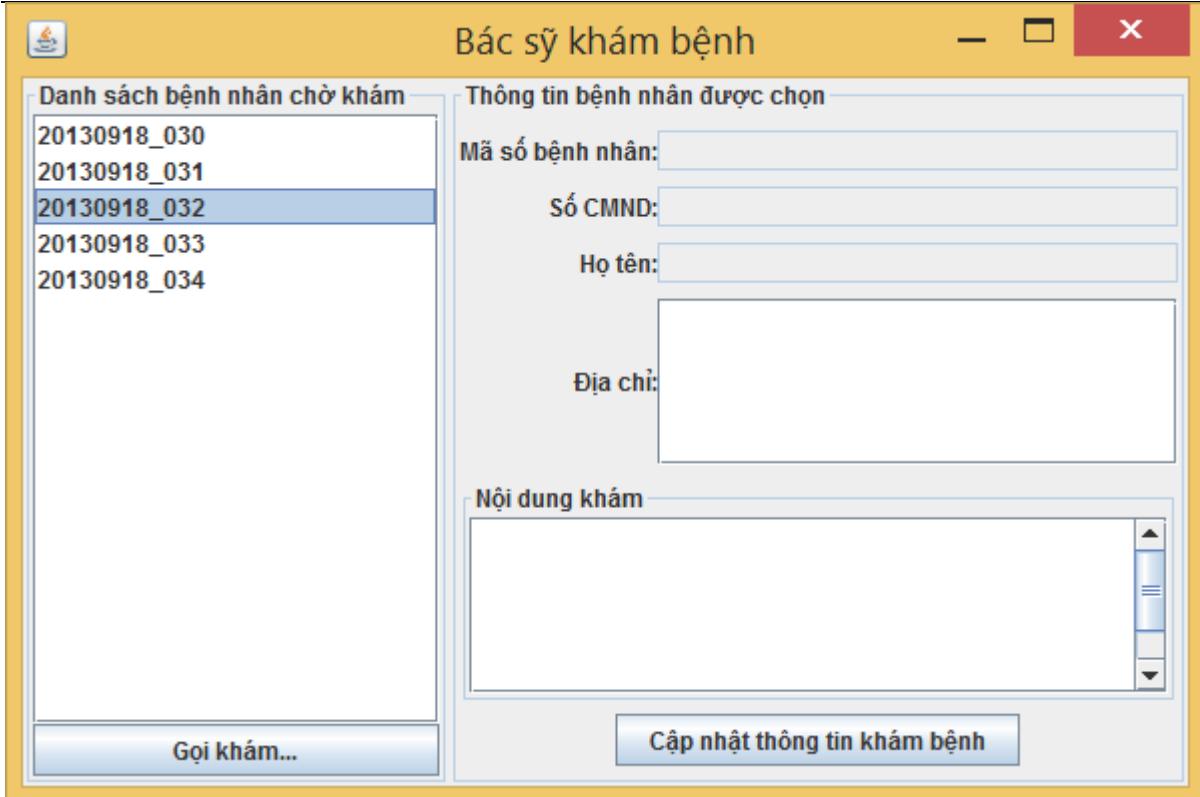
Tại máy tính của Nhân viên lễ tân (viết bằng C#):



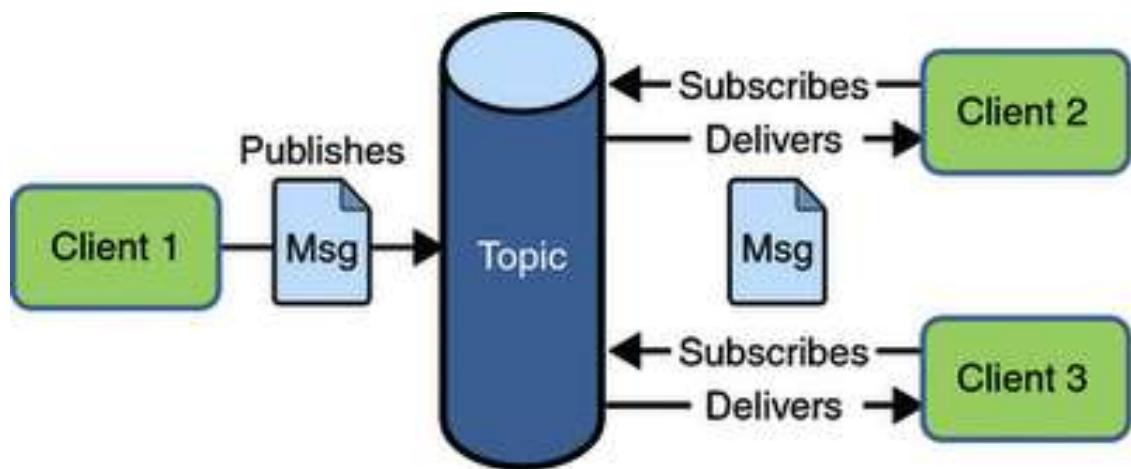
Khi bệnh nhân vào khám thì nhân viên lễ tân sẽ nhập thông tin của bệnh nhân nếu đến khám lần đầu hoặc tìm thông tin bệnh nhân nếu tái khám (tìm theo mã số bệnh nhân hoặc theo số CMND). Sau khi tìm thấy hoặc nhập vào, nhấn nút “Lưu thông tin” thì chương trình sẽ notify cho máy tính đặc trong phòng của bác sĩ khám (giả sử có n bác sĩ, mỗi bác sĩ có một máy chạy độc lập).

Tại máy tính trong phòng bác sĩ (viết bằng Java)

Khi có 1 bệnh nhân cần khám, chương trình sẽ đưa vào danh sách đợi để bác sĩ gọi vào.



2. Giả sử ta có một công việc cần phân chia cho N máy tính làm riêng rẽ sau đó ta tổng hợp kết quả lại. Yêu cầu mô phỏng cho hệ thống này: lập tài liệu thiết kế, hiện thực chương trình.

MÔ HÌNH PUBLISHER – SUBSCRIBER**CODE MINH HỌA CHO GỬI VÀ NHẬN BẰNG JAVA****Publisher**

```

import java.util.Properties;
import javax.jms.Connection;
import javax.jms.ConnectionFactory;
import javax.jms.Destination;
import javax.jms.Message;
import javax.jms.MessageProducer;
import javax.jms.Session;
import javax.naming.Context;
import javax.naming.InitialContext;
import org.apache.log4j.BasicConfigurator;

public class TopicPublisher {
    public static void main(String[] args) throws Exception{
        //thiết lập môi trường cho JMS logging
        BasicConfigurator.configure();
        //thiết lập môi trường cho JJNDI
        Properties settings=new Properties();
        settings.setProperty(Context.INITIAL_CONTEXT_FACTORY,
                            "org.apache.activemq.jndi.ActiveMQInitialContextFactory");
        settings.setProperty(Context.PROVIDER_URL, "tcp://localhost:61616");
        //tạo context
        Context ctx=new InitialContext(settings);
        //lookup JMS connection factory
        Object obj=ctx.lookup("TopicConnectionFactory");
        ConnectionFactory factory=(ConnectionFactory)obj;
        //tạo connection
        Connection con=factory.createConnection("admin","admin");
        //nối đến MOM
        con.start();
        //tạo session
        Session session=con.createSession(
            /*transaction*/false,
            /*ACK*/Session.AUTO_ACKNOWLEDGE
        );
        Destination destination=(Destination)
        ctx.lookup("dynamicTopics/thanthidet");

        //tạo producer
        MessageProducer producer = session.createProducer(destination);
        //Tạo 1 message
    }
}

```

```

        Message msg=session.createTextMessage("xin chào người ẹp");

        //gửi
        producer.send(msg);
        //shutdown connection
        session.close();
        con.close();

        System.out.println("Finished...");
    }
}

```

Subscriber

```

import java.util.Properties;
import javax.jms.Connection;
import javax.jms.ConnectionFactory;
import javax.jms.Destination;
import javax.jms.Message;
import javax.jms.MessageConsumer;
import javax.jms.MessageListener;
import javax.jms.Session;
import javax.jms.TextMessage;
import javax.naming.Context;
import javax.naming.InitialContext;
import org.apache.log4j.BasicConfigurator;

public class TopicSubscriber {
    public static void main(String[] args) throws Exception{
        //thiết lập môi trường cho JMS
        BasicConfigurator.configure();
        //thiết lập môi trường cho JNDI
        Properties settings=new Properties();
        settings.setProperty(Context.INITIAL_CONTEXT_FACTORY,
                            "org.apache.activemq.jndi.ActiveMQInitialContextFactory");
        settings.setProperty(Context.PROVIDER_URL, "tcp://localhost:61616");
        //tạo context
        Context ctx=new InitialContext(settings);
        //lookup JMS connection factory
        Object obj=ctx.lookup("TopicConnectionFactory");
        ConnectionFactory factory=(ConnectionFactory)obj;
        //tạo connection
        Connection con=factory.createConnection("admin","admin");
        //nối đến MOM
        con.start();
        //tạo session
        Session session=con.createSession(
            /*transaction*/false,
            /*ACK*/Session.CLIENT_ACKNOWLEDGE
        );
        //tạo consumer
        Destination destination=(Destination) ctx.lookup("dynamicTopics/thanthidet");
        MessageConsumer receiver = session.createConsumer(destination);
        //receiver.receive(); //blocked method
        //Cho receiver lắng nghe trên queue, chừng có message thì notify
        receiver.setMessageListener(new MessageListener() {
            @Override
            //có message đến queue, phương thức này được thực thi
            public void onMessage(Message msg) { //msg là message nhận được
                try {
                    if(msg instanceof TextMessage){
                        TextMessage tm=(TextMessage)msg;

```

```
        String txt=tm.getText();
        System.out.println("XML= "+txt);
        msg.acknowledge(); //gửi tín hiệu ack
    }
} catch (Exception e) {
    e.printStackTrace();
}
});
```

CODE MINH HỌA CHO GỬI VÀ NHẬN BẰNG C#

Publisher

```
using System;
using Apache.NMS;
using Apache.NMS.ActiveMQ;
using Apache.NMS.ActiveMQ.Commands;
namespace TopicSender
{
    class Program
    {
        static void Main(string[] args)
        {
            IConnectionFactory factory = new ConnectionFactory("tcp://localhost:61616");
            IConnection con = factory.CreateConnection("admin", "admin");
            con.Start(); //nối tới MOM
            ISession session = con.CreateSession(AcknowledgementMode.AutoAcknowledge);

            ActiveMQTopic destination = new ActiveMQTopic("thanthidet");
            IMessageProducer producer = session.CreateProducer(destination);
            ActiveMQTextMessage tmsg = new ActiveMQTextMessage("hola. mondo");
            producer.Send(tmsg);
            Console.WriteLine("press any key to continue");
            Console.ReadKey();
        }
    }
}
```

Subscriber

```
using System;
using Apache.NMS;
using Apache.NMS.ActiveMQ;
using Apache.NMS.ActiveMQ.Commands;
namespace TopicReceiver
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("press enter to exit");
            IConnectionFactory factory = new ConnectionFactory("tcp://localhost:61616");
            IConnection con = factory.CreateConnection("admin", "admin");
            con.Start(); //nối tới MOM
            ISession session = con.CreateSession(AcknowledgementMode.AutoAcknowledge);

            ActiveMQTopic destination = new ActiveMQTopic("thanthidet");
            IMessageConsumer producer = session.CreateConsumer(destination);

            producer.Listener += ProducerListener;
        }
    }
}
```

```

        Console.ReadLine();
    }

    private static void Producer_Listener(IMessage message)
    {
        if(message is ActiveMQTextMessage)
        {
            ActiveMQTextMessage tmsg = message as ActiveMQTextMessage;
            String txt = tmsg.Text;
            Console.WriteLine(txt);
        }
    }
}

```

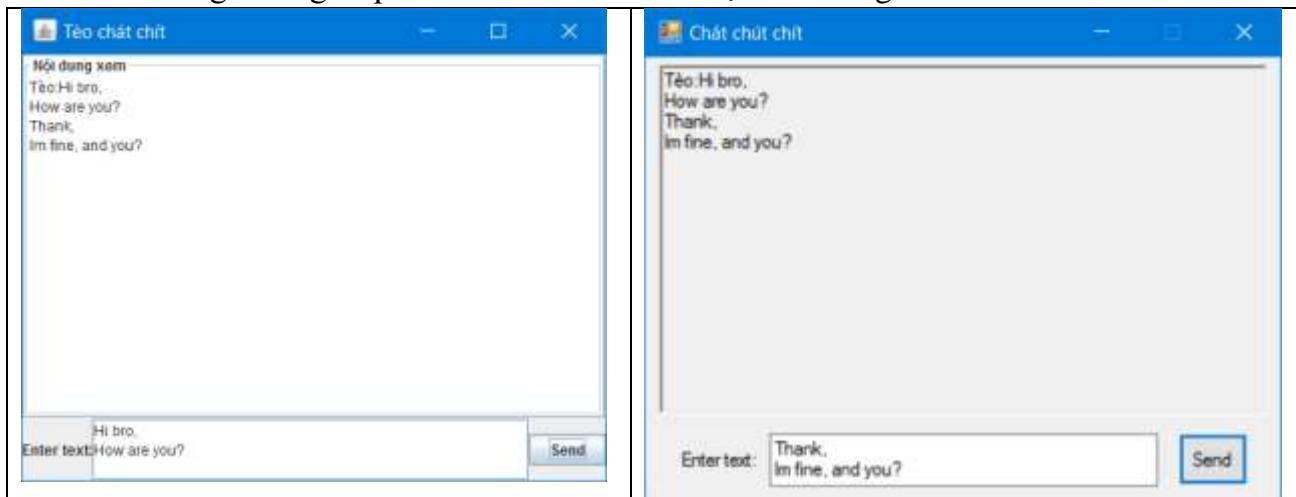
THỰC THI CHÉO

1. Chạy Sender của Java, Receiver của C#
2. Chạy Sender của C#, Receiver của Java

Quan sát kết quả

BÀI TẬP

1. Viết chương trình groups chat với chat client được viết bằng Java và C#



2. *** Viết chương trình bằng java thực hiện công việc như sau:

Chương trình chia sẻ màn hình giữa một máy server và một tập các máy client.

- Máy server có thể trong thấy màn hình của tất cả các máy clients kết nối
- Khi máy server chọn chế độ trình diễn thì tất cả các máy client sẽ thấy được màn hình chia sẻ của máy server.

Hãy viết tài liệu thiết kế sau đó cài đặt hệ thống này

SECURITY**CẤU HÌNH****Authetication****Simple Authentication plugin**

Những gì bạn cần làm để thêm các groups và các users là mở file activemq.xml lên và thêm phân đoạn sau vào trong thẻ plugins

```
<plugins>
    <!-- Configure authentication: Username, passwords and groups -->
    <simpleAuthenticationPlugin anonymousAccessAllowed="true">
        <users>
            <authenticationUser username="system" password="manager" groups="users,admins"/>
            <authenticationUser username="user" password="password" groups="users"/>
            <authenticationUser username="guest" password="password" groups="guests"/>
        </users>
    </simpleAuthenticationPlugin>
```

JAAS Authetication Plugin

Mở file %ACTIVEMQ_HOME%\conf\login.config, hiệu chỉnh file sẽ lưu user và groups như hình

```
activemq {
    org.apache.activemq.jaas.PropertiesLoginModule required
    org.apache.activemq.jaas.properties.user="users.properties"
    org.apache.activemq.jaas.properties.group="groups.properties"
    reload=true;
};
```

Ta có thể tạo nhiều configuration khác nhau. Ở trên là một cấu hình có tên mặc định activemq.

Sau đó ta thêm một số nhóm và một số user thuộc nhóm.

```
#group_name=users_list
admins=admin,teo
users=ty
dhcn=men
guests=guest
```

groups.properties
Mỗi dòng chỉ một tên nhóm gồm những user nào

```
#username=password
admin=admin
teo=123
ty=456
men=123
```

users.properties
mỗi dòng có dạng: tên_user=mật_khẩu

Thêm phân đoạn sau vào trong file activemq.xml trong thẻ plugins

```
<jaasAuthenticationPlugin configuration="activemq" />
```

AUTHORIZATION

Trong ActiveMQ, ta có thể gán quyền cho các groups (tên hoặc các ký tự thay thế (wild-card)) bằng một trong các quyền sau

read	Có thể browse và consume từ một destination (queue/topic)
write	Có thể gửi message đến một destination
admin	Quyền admin

Xem ví dụ ở phân đoạn sau:

```

<!-- Lets configure a destination based authorization mechanism -->
<authorizationPlugin>
    <map>
        <authorizationMap>
            <authorizationEntries>
                <authorizationEntry queue=">" read="admins" write="admins" admin="admins" />
                <authorizationEntry queue="USERS.>" read="users" write="users" admin="users" />
                <authorizationEntry queue="GUEST.>" read="guests" write="guests,users"
                    admin="guests,users" />
                <authorizationEntry queue="TEST.Q" read="guests" write="guests" />
                <authorizationEntry topic=">" read="admins" write="admins" admin="admins" />
                <authorizationEntry topic="USERS.>" read="users" write="users" admin="users" />
                <authorizationEntry topic="GUEST.>" read="guests" write="guests,users"
                    admin="guests,users" />
                <authorizationEntry topic="ActiveMQ.Advisory.>" read="guests,users"
                    write="guests,users" admin="guests,users"/>
            </authorizationEntries>
            <tempDestinationAuthorizationEntry>
                <tempDestinationAuthorizationEntry read="admin" write="admin" admin="admin"/>
            </tempDestinationAuthorizationEntry>
        </authorizationMap>
    </map>
</authorizationPlugin>

```

Ví dụ:

```
<authorizationEntry queue=">" read="admins" write="admins" admin="admins" />
```

điều đó có nghĩa: những users thuộc group admins có thể toàn quyền nhận, gửi và quản lý trên tất cả các queue có trong hệ thống.

Trong quá trình tạo ra một Connection, ta phải cung cấp 2 tham số **<username, password>** để có thể đăng nhập vào destination

```

Properties settings=new Properties();
settings.setProperty(Context.INITIAL_CONTEXT_FACTORY,
    "org.apache.activemq.jndi.ActiveMQInitialContextFactory");
settings.setProperty(Context.PROVIDER_URL, "tcp://localhost:61616");
//create context
Context ctx=new InitialContext(settings);
//lookup JMS connection factory
Object obj=ctx.lookup("ConnectionFactory");
ConnectionFactory factory=(ConnectionFactory)obj;
//create connection
Connection con=factory.createConnection("admin","admin");
//connect to MOM
con.start();

```

Nếu bạn cung cấp sai thông tin username-password thì bạn không thể authenticate

Nếu bạn đã authenticate mà không được cấp quyền trên destination thì bạn không được authorize

BÀI TẬP

1. Với bài tập 1 phần mô hình P2P,
 - Tạo 2 users có tên **LanHT** và **HungNV** với mật khẩu tương ứng là 123Cba và 321Abc thuộc vào 2 nhóm có tên **bacsy** và **letan**.
 - Cấu hình security sao cho nhân viên thuộc nhóm letan chỉ được phép gửi và nhân viên thuộc nhóm bacsy chỉ có quyền đọc queue **KhamBenh**.
2. Với bài 2 phần mô hình publisher-subscriber,
 - Tạo 3 nhóm có tên và mục đích như sau:
 - Teacher – dùng cho máy server publish, các máy client subscribe
 - Student – dùng cho các máy client publisher và máy server subscribe
 - Tạo Topic Presentation chỉ cho nhóm Teacher write, admin
 - Tạo

- Câu hình sao cho các máy nhóm client chỉ được quyền read cho topic có tên trình chiếu

MULTITIERS ARCHITECTURE

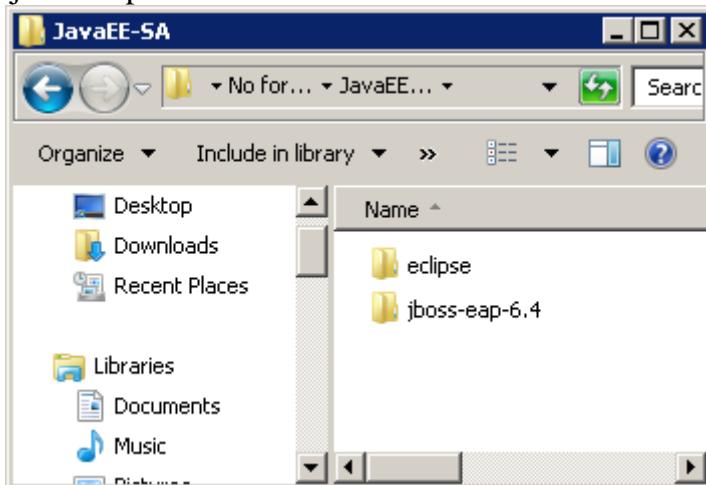
I. HƯỚNG DẪN LÀM VIỆC VỚI JBOSS-ECLIPSE-JAVAEE

CÀI ĐẶT VÀ CẤU HÌNH

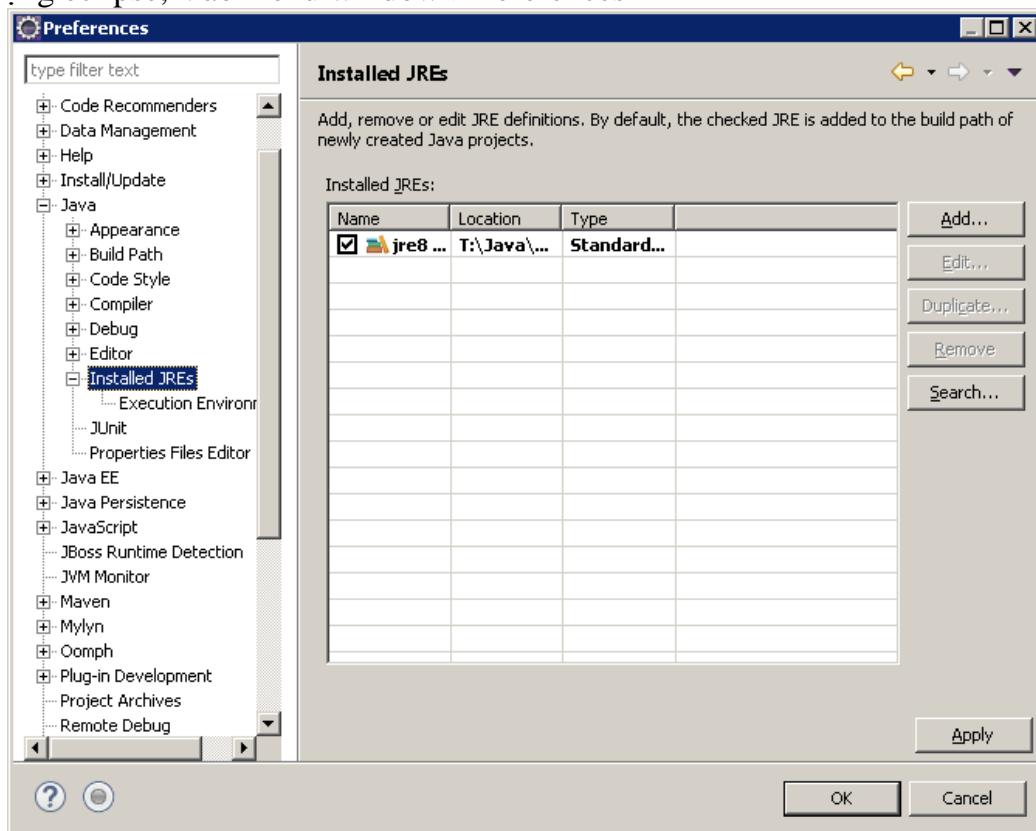
1. Cài đặt JDK 8. Nên cài vào nơi khác Program File
2. Thiết lập biến môi trường JAVA_HOME



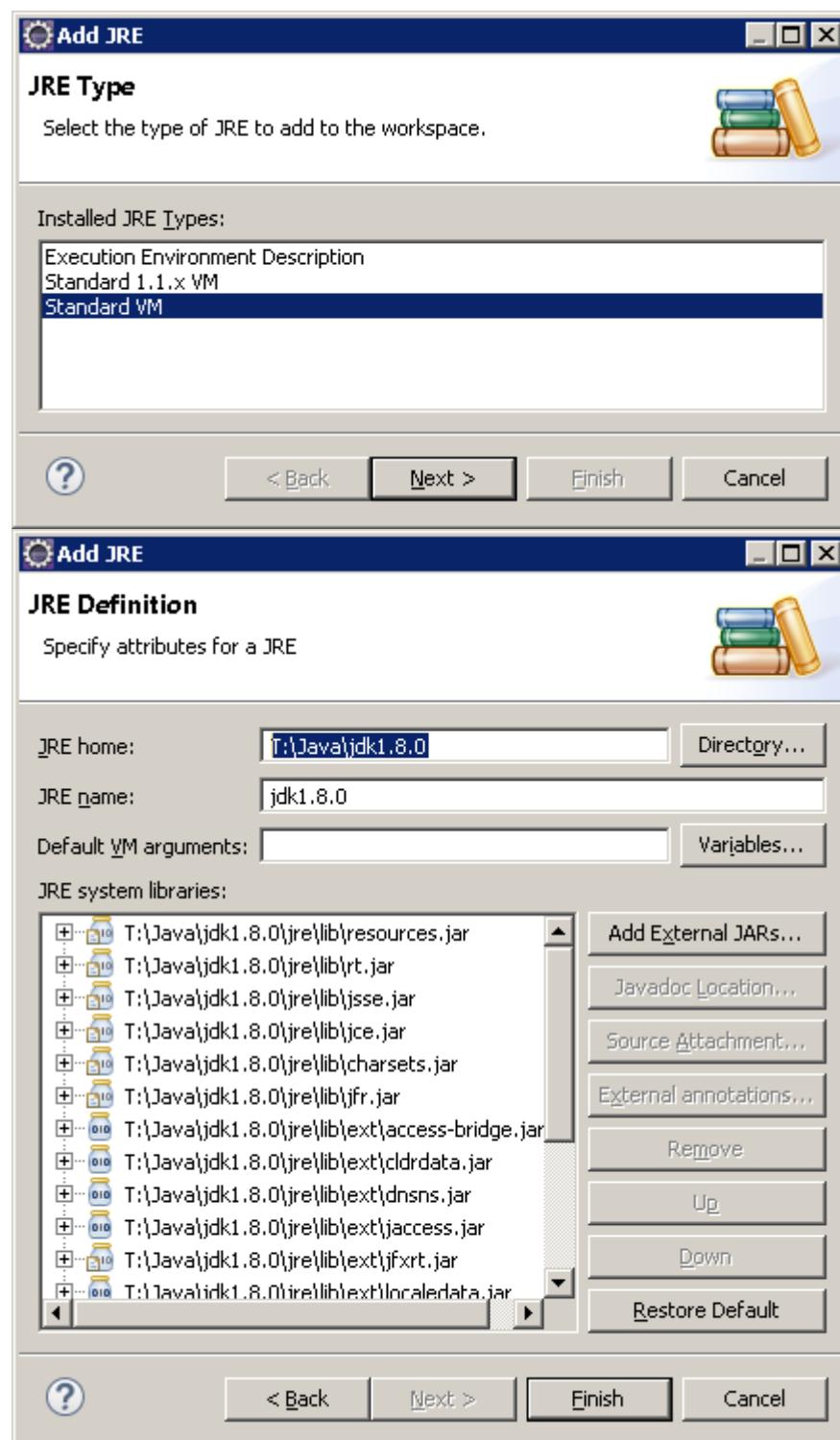
3. Giải nén eclipse và jboss-eap

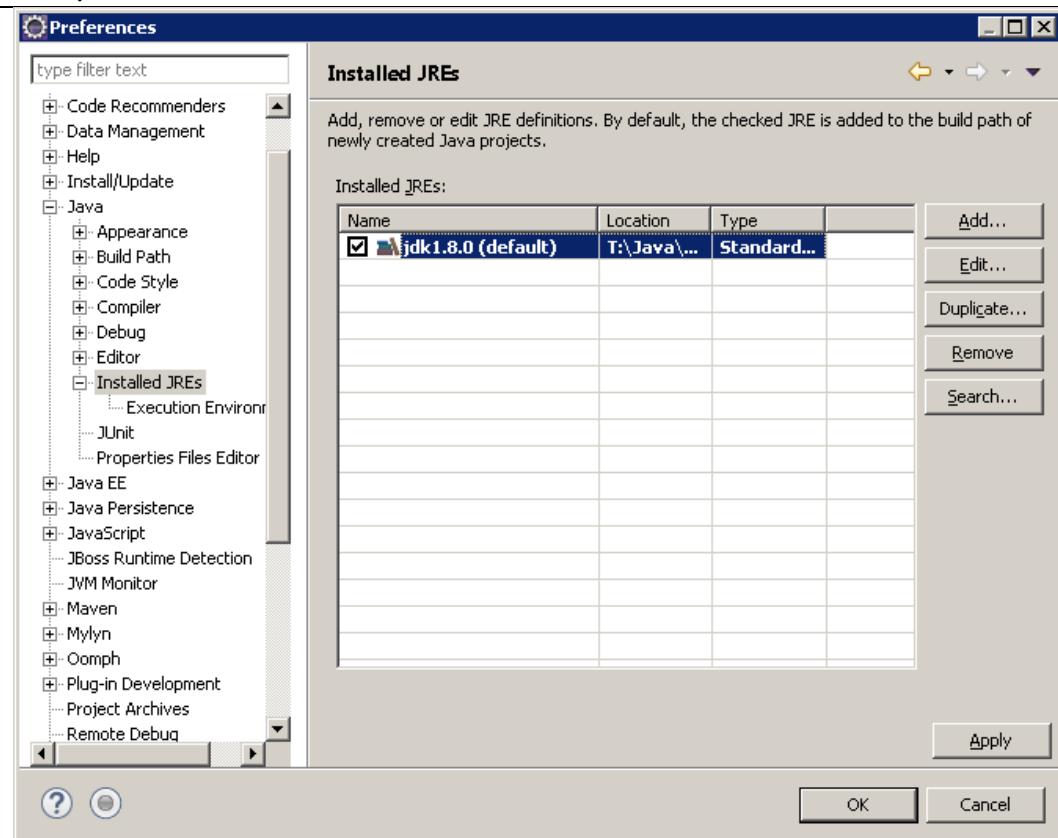


4. Khởi động eclipse, Vào menu window->References



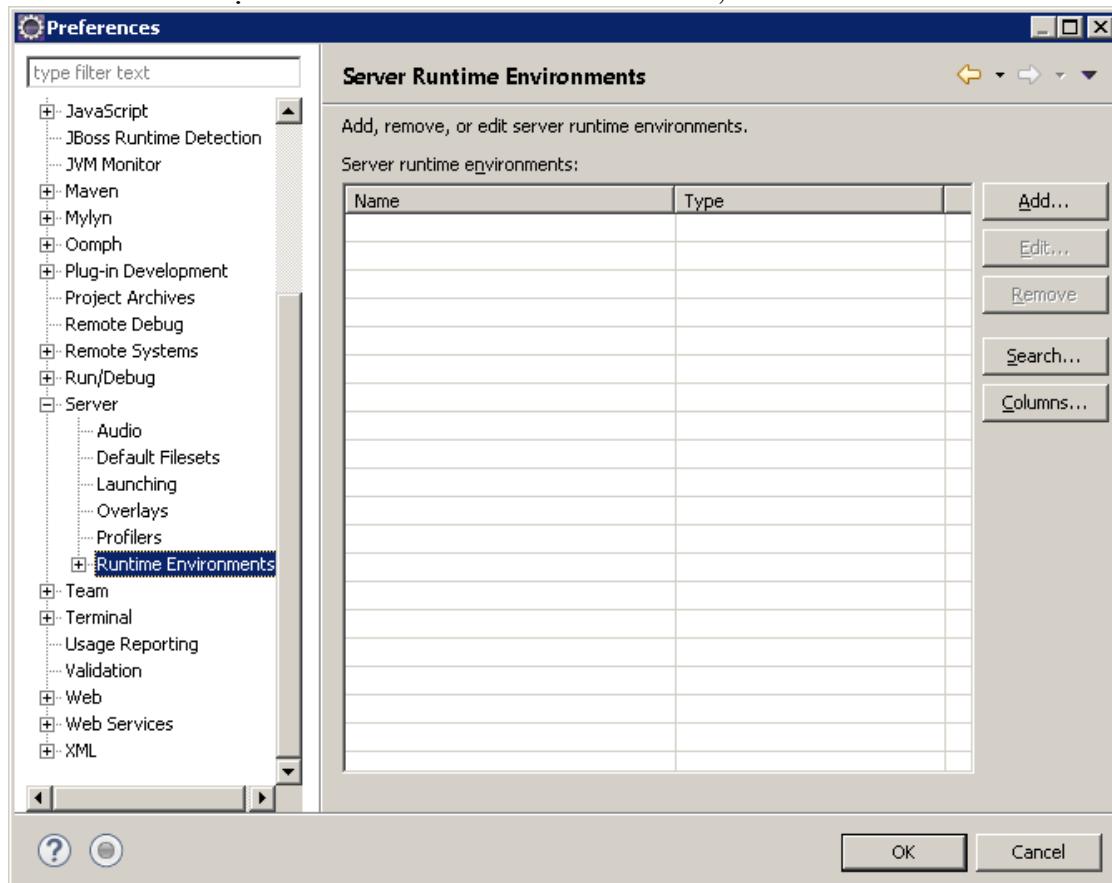
Chọn Java->Installed JREs, them Runtime(nhấn nút Add)

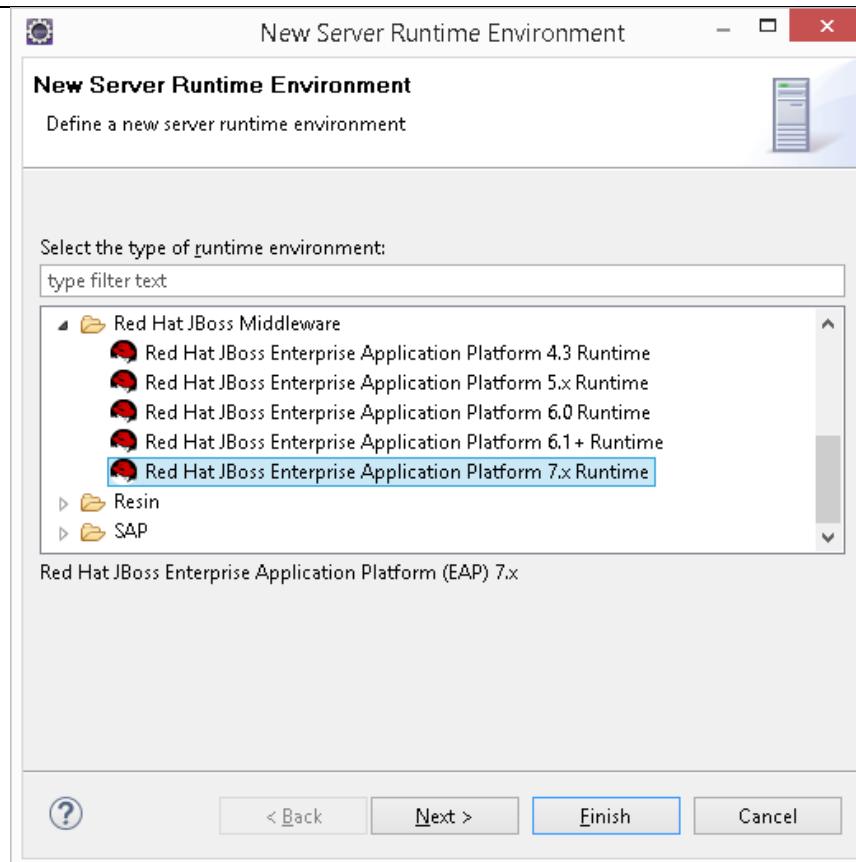




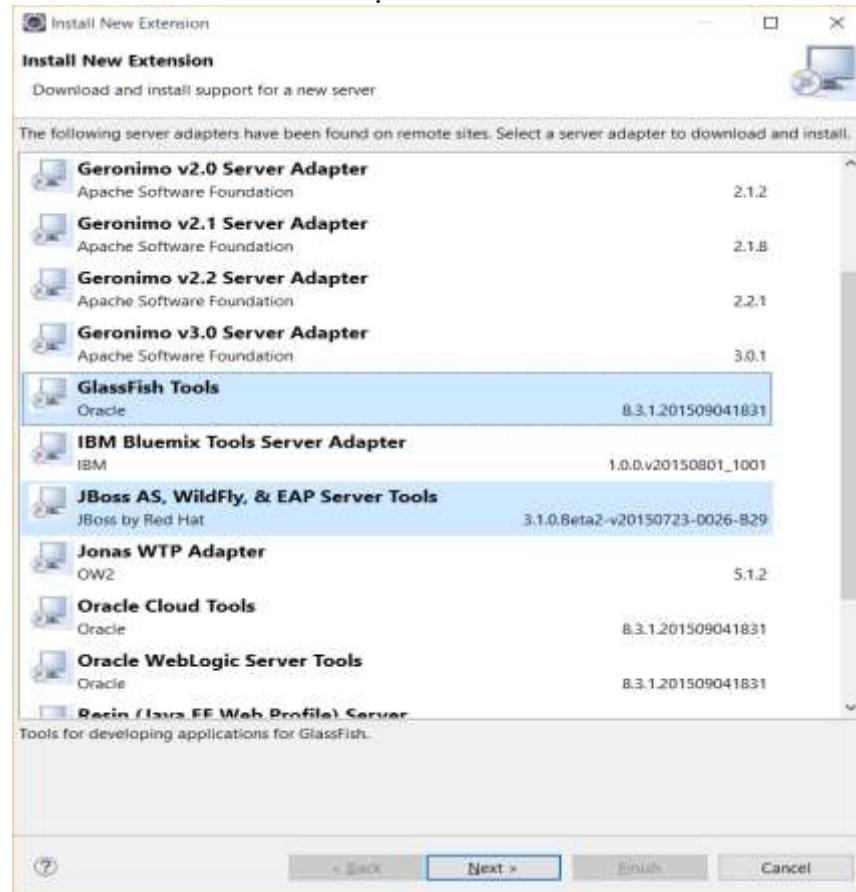
Nhấn Apply

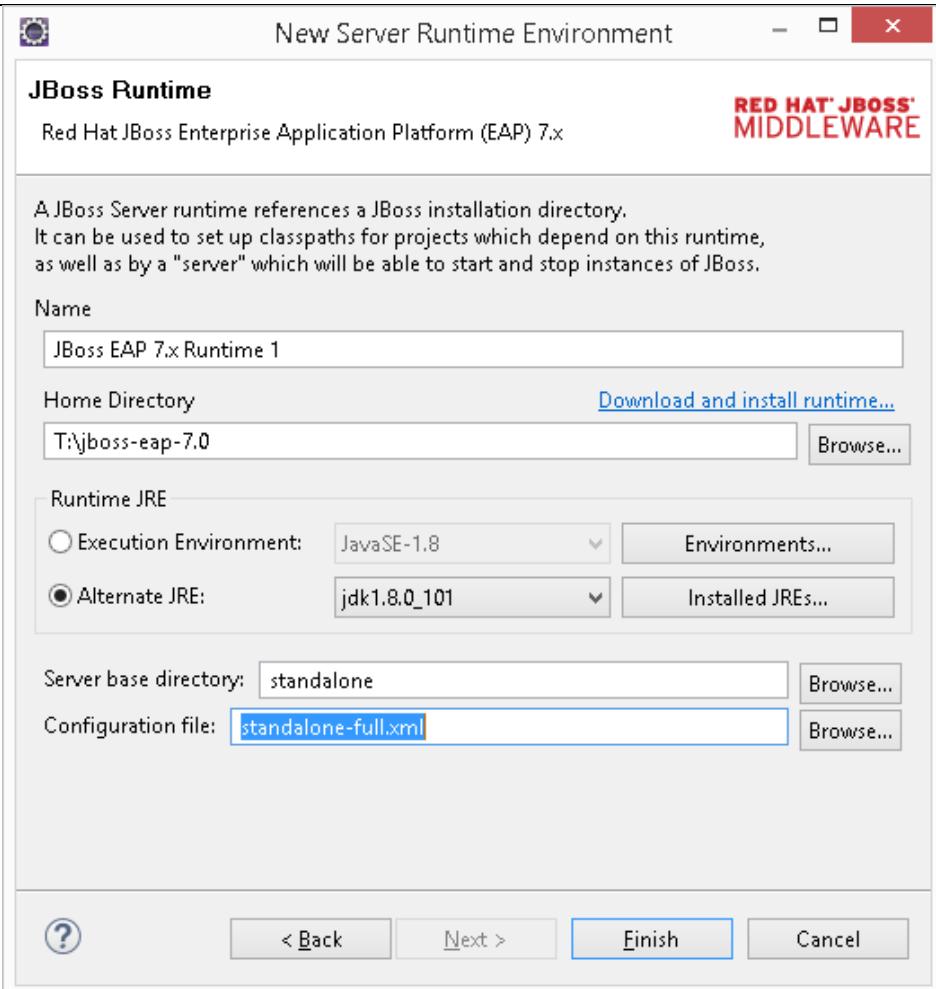
5. Cấu hình Server: Chọn Server->Runtime Environments, Nhấn nút Add





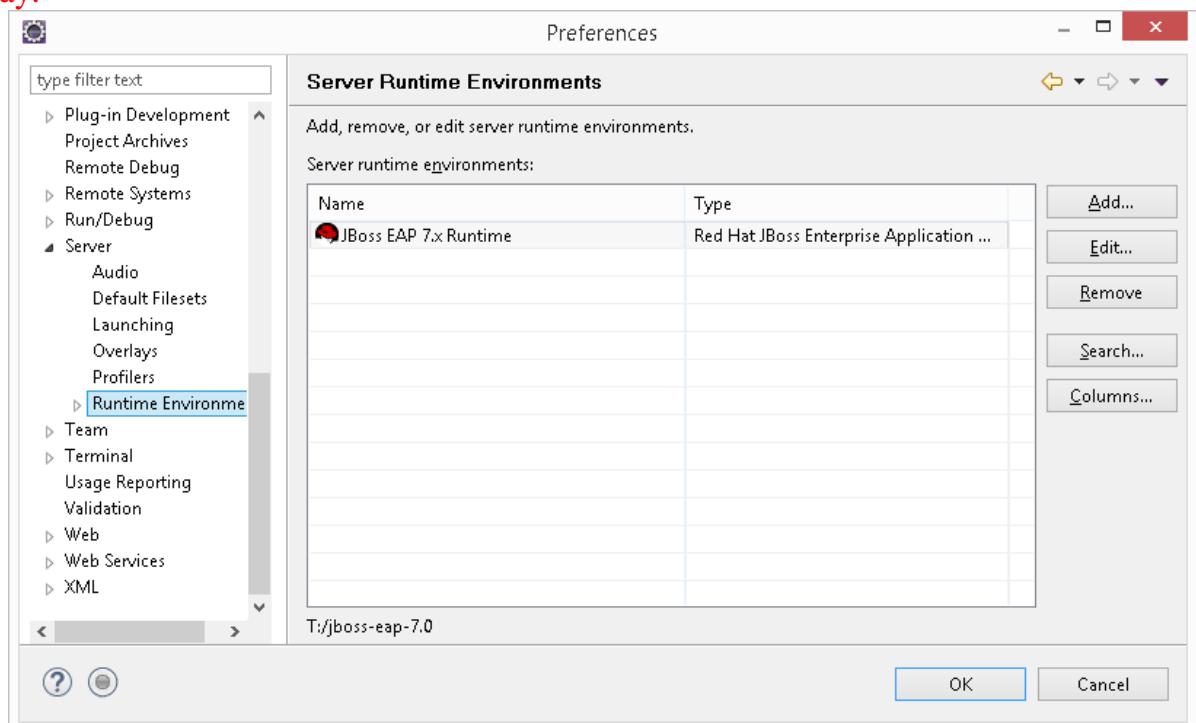
Trong trường hợp không tìm thấy Server Runtime Environment, bạn nhấp vào link “Show additional server adapter” (*yêu cầu có kết nối với internet*) bạn sẽ thấy 1 danh sách các RE khác nhau. Tìm đến Jboss AS để chọn như hình





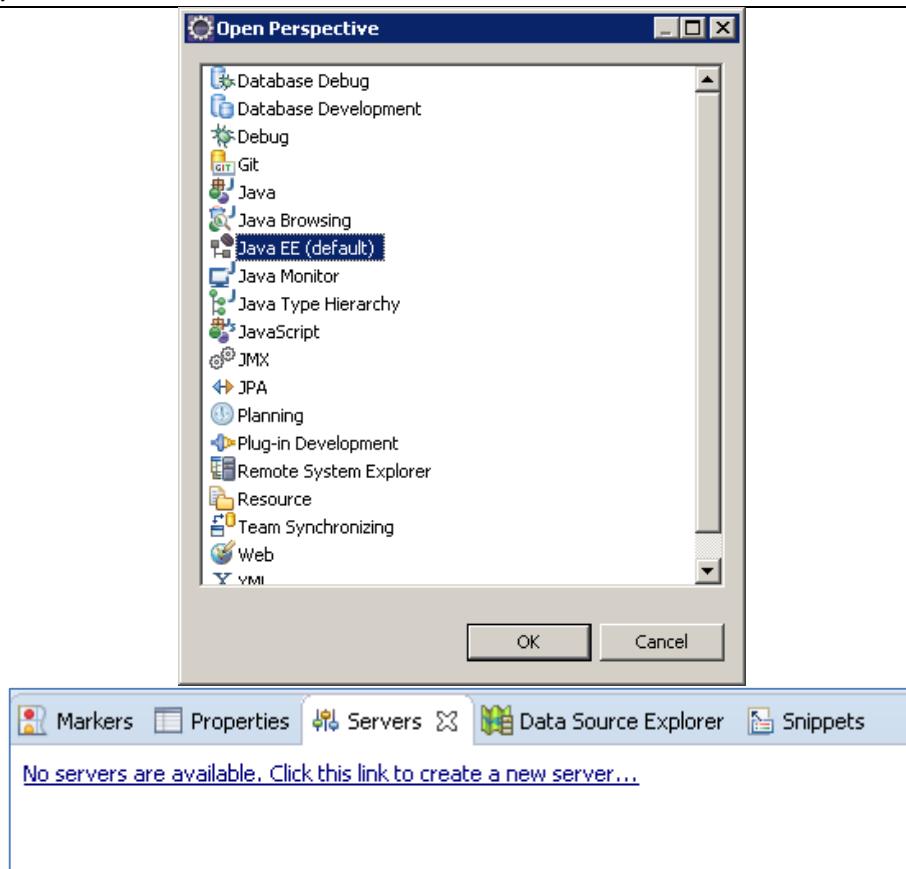
Home directory: chọn đường dẫn đến thư mục giải nén JBoss EAP

Configuration file: **standalone-full.xml** để sau này cấu hình security và các phần khác sau này.

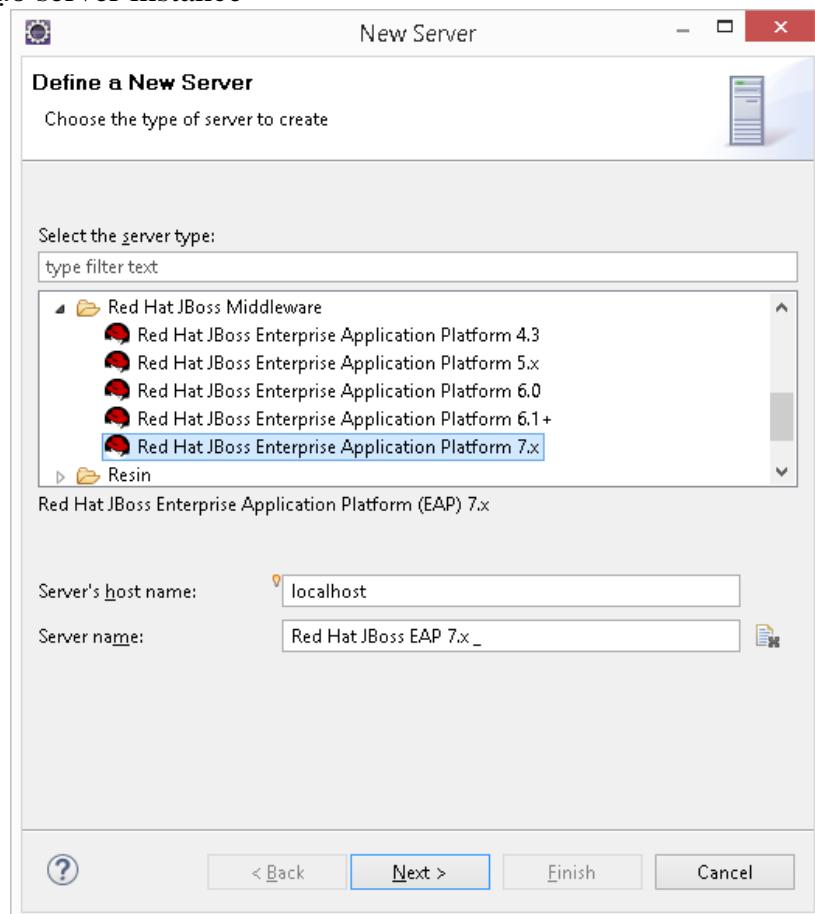


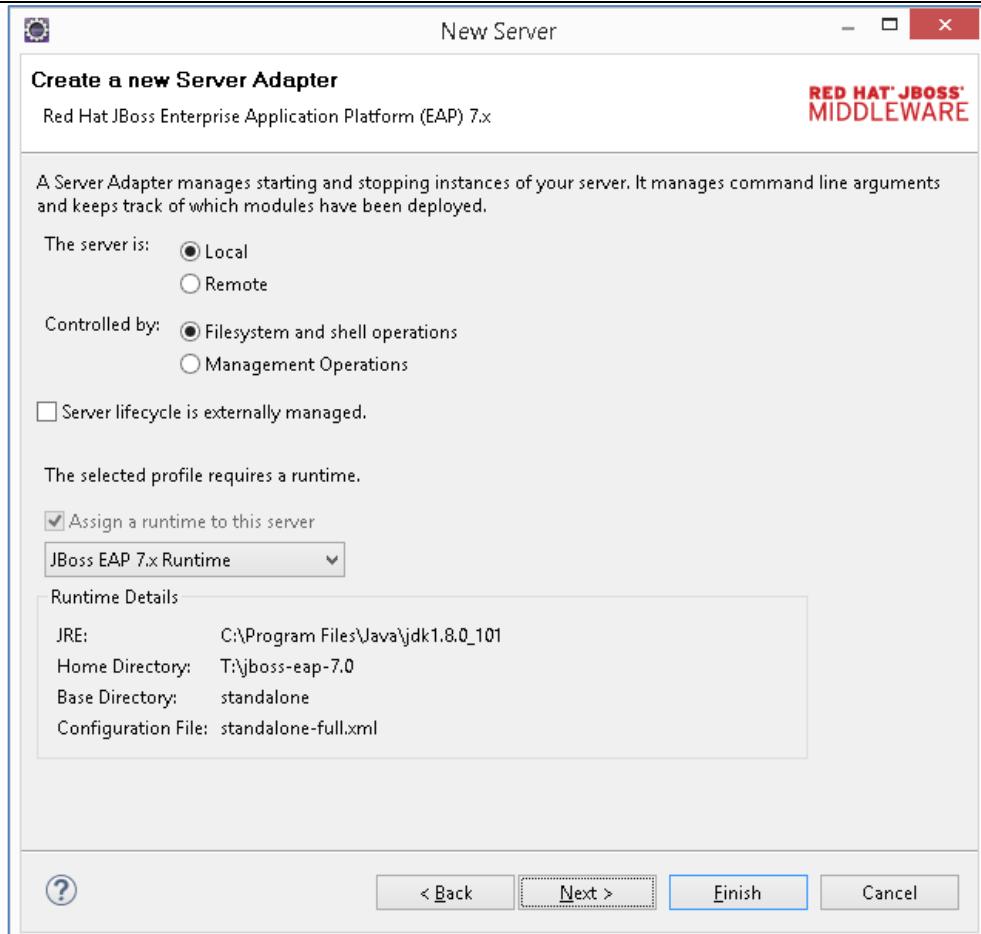
6. Cấu hình perspective

Chọn perspective Java EE

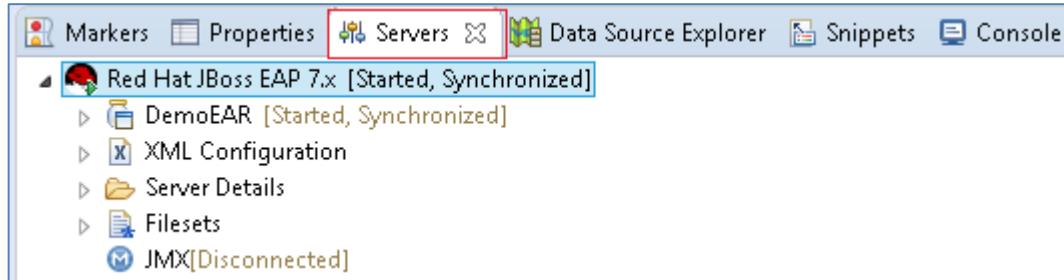


Nhấn link để tạo server instance

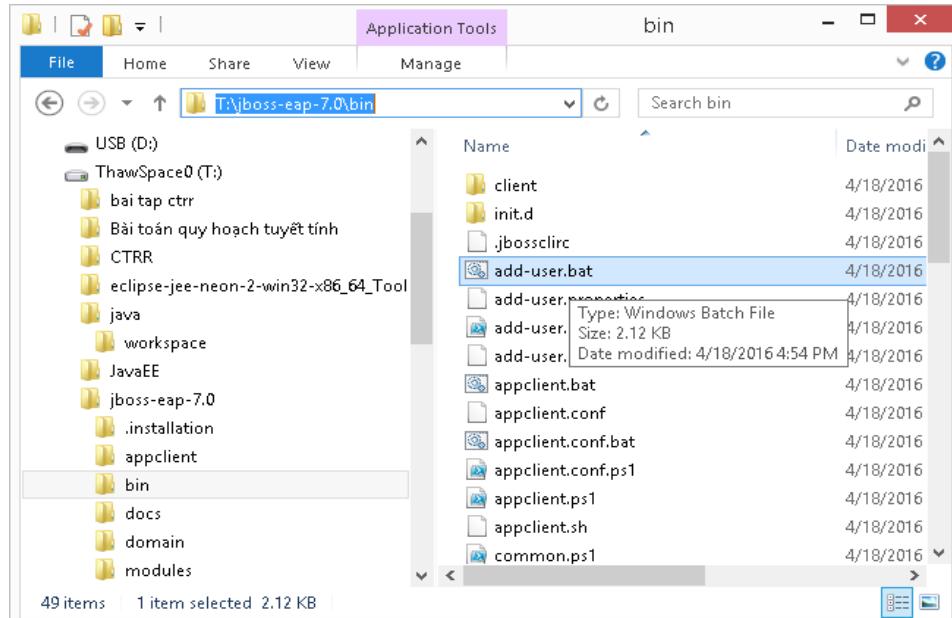




Nhấn Finish



7. Cấu hình user cho JbossEAP



```

C:\Windows\system32\cmd.exe
What type of user do you wish to add?
  a) Management User <mgmt-users.properties>
  b) Application User <application-users.properties>
(a): a

C:\Windows\system32\cmd.exe
What type of user do you wish to add?
  a) Management User <mgmt-users.properties>
  b) Application User <application-users.properties>
(a): a

Enter the details of the new user to add.
Using realm 'ManagementRealm' as discovered from the existing configuration file.
Username : men
Password requirements are listed below. To modify these restrictions, edit the <user.properties> configuration file.
- The password must not be one of the following restricted administrator accounts.
- The password must contain at least 8 characters, 1 alpha digit(s), 1 non-alphanumeric symbol(s).
- The password must be different from the username.
Password : -

```

Password: Abc123!@# hoặc pass khác đảm bảo tính security

```

C:\Windows\system32\cmd.exe
(a): a

Enter the details of the new user to add.
Using realm 'ManagementRealm' as discovered from the existing configuration file.
Username : men
Password requirements are listed below. To modify these restrictions, edit the <user.properties> configuration file.
- The password must not be one of the following restricted administrator accounts.
- The password must contain at least 8 characters, 1 alpha digit(s), 1 non-alphanumeric symbol(s).
- The password must be different from the username.
Password :
Re-enter Password :
What groups do you want this user to belong to? (Please enter a list, or leave blank for none)[ ]: management
About to add user 'men' for realm 'ManagementRealm'
Is this correct yes/no?

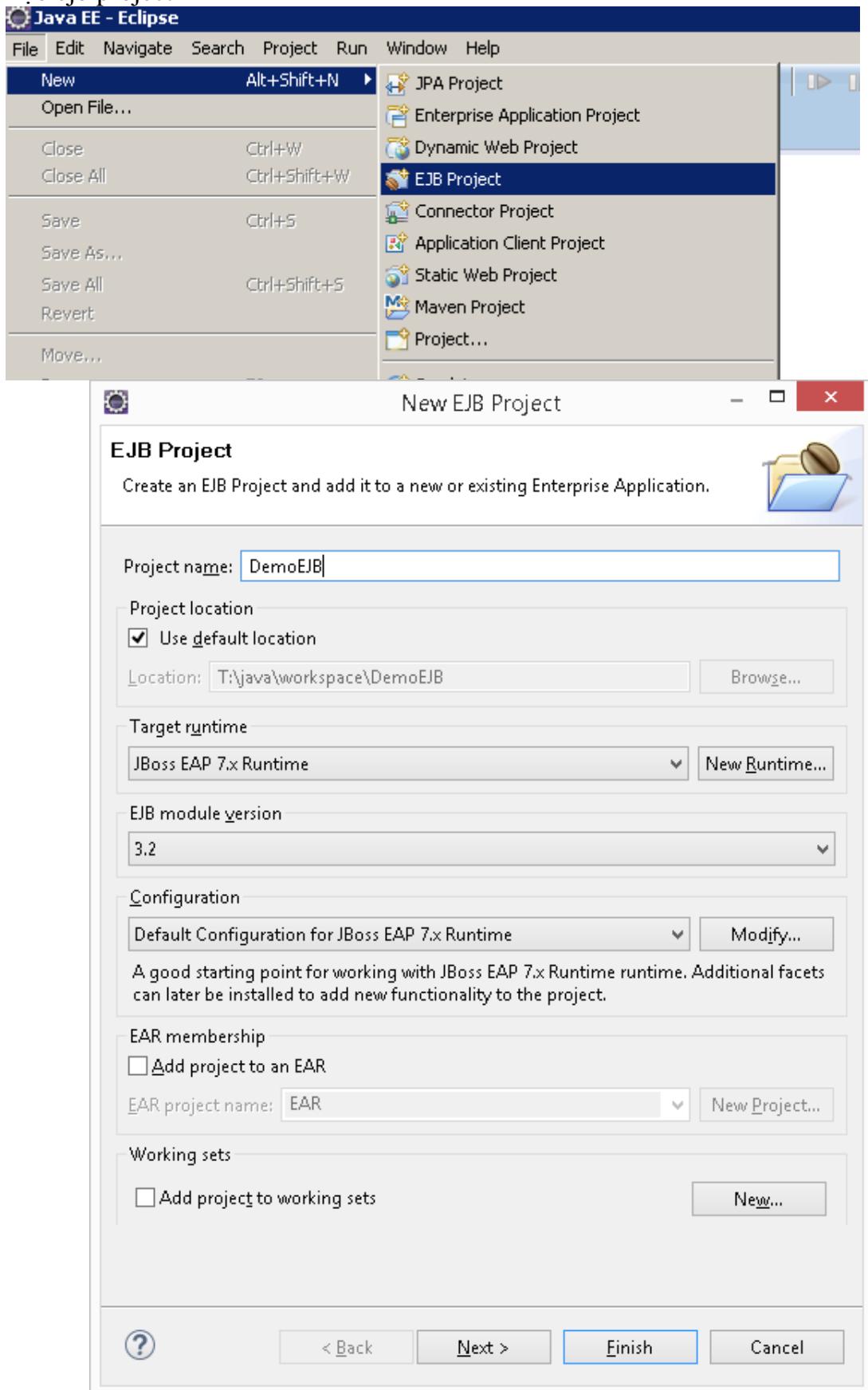
```

Chọn yes 2 lần để kết thúc

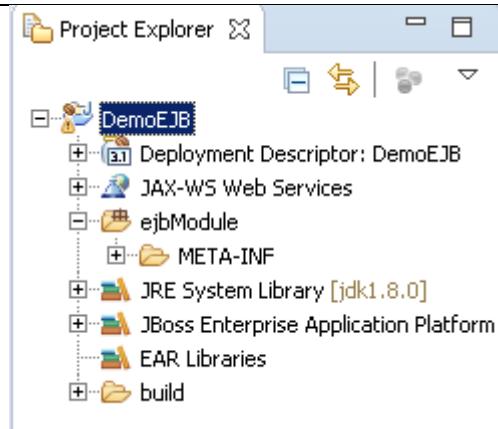
II. STATELESS SESSION BEAN

EJB PROJECT

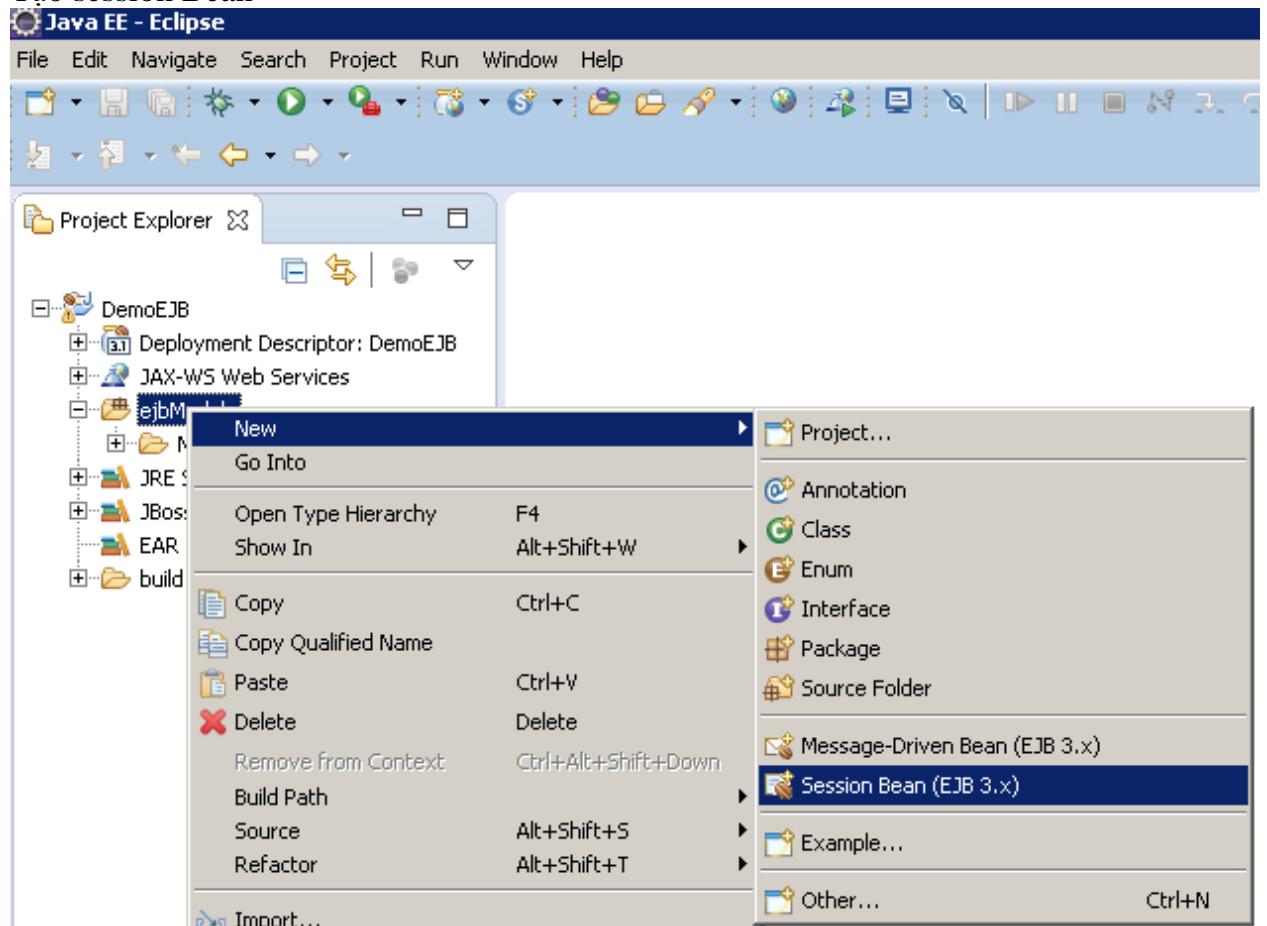
1. Tạo ejb project

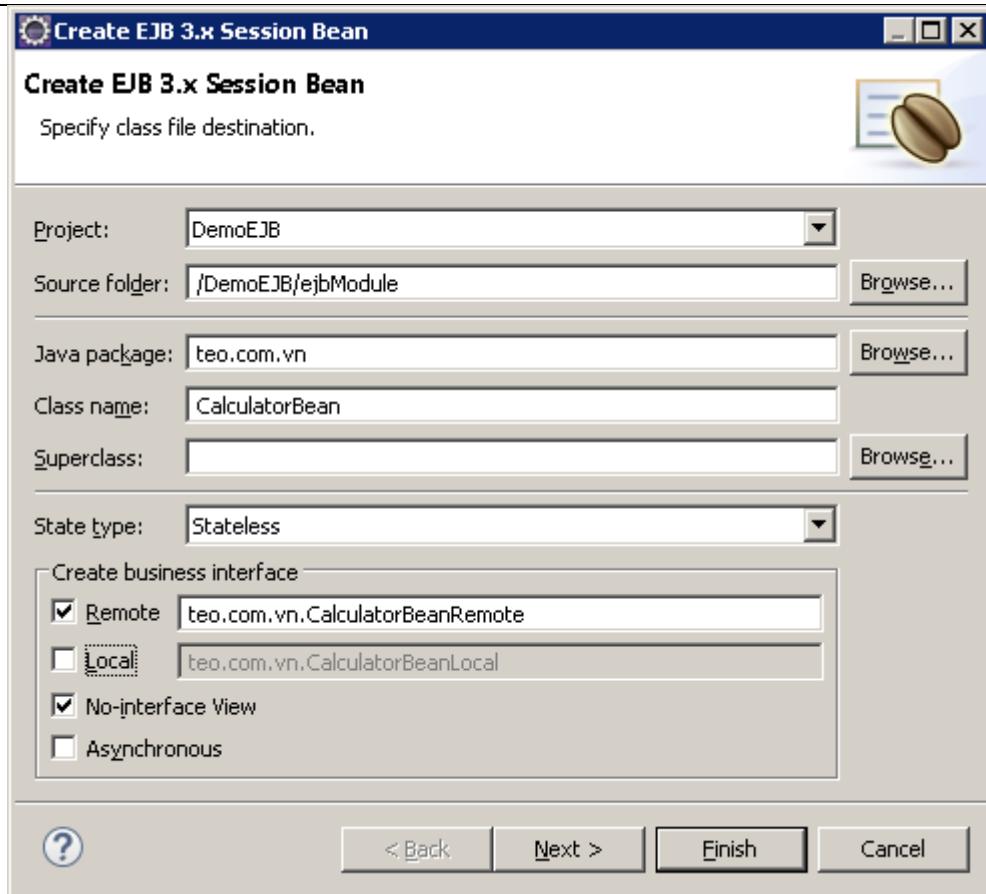


Nhấn Finish

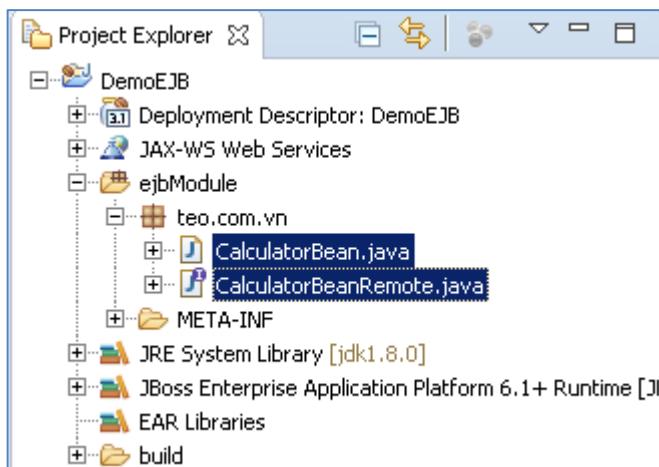


2. Tạo session Bean





Nhấn Finish



3. Viết logic

Chú ý: những phương thức nào mà client có thể triệu gọi từ xa được phải nằm trong remote interface

```

1 package teo.com.vn;
2
3 import javax.ejb.Remote;
4
5 @Remote
6 public interface CalculatorBeanRemote {
7     public abstract int cong(int a,int b);
8     int tru(int a,int b); //mac định là public abstract
9     int nhan(int a,int b);
10    int chia(int a,int b);
11 }

```

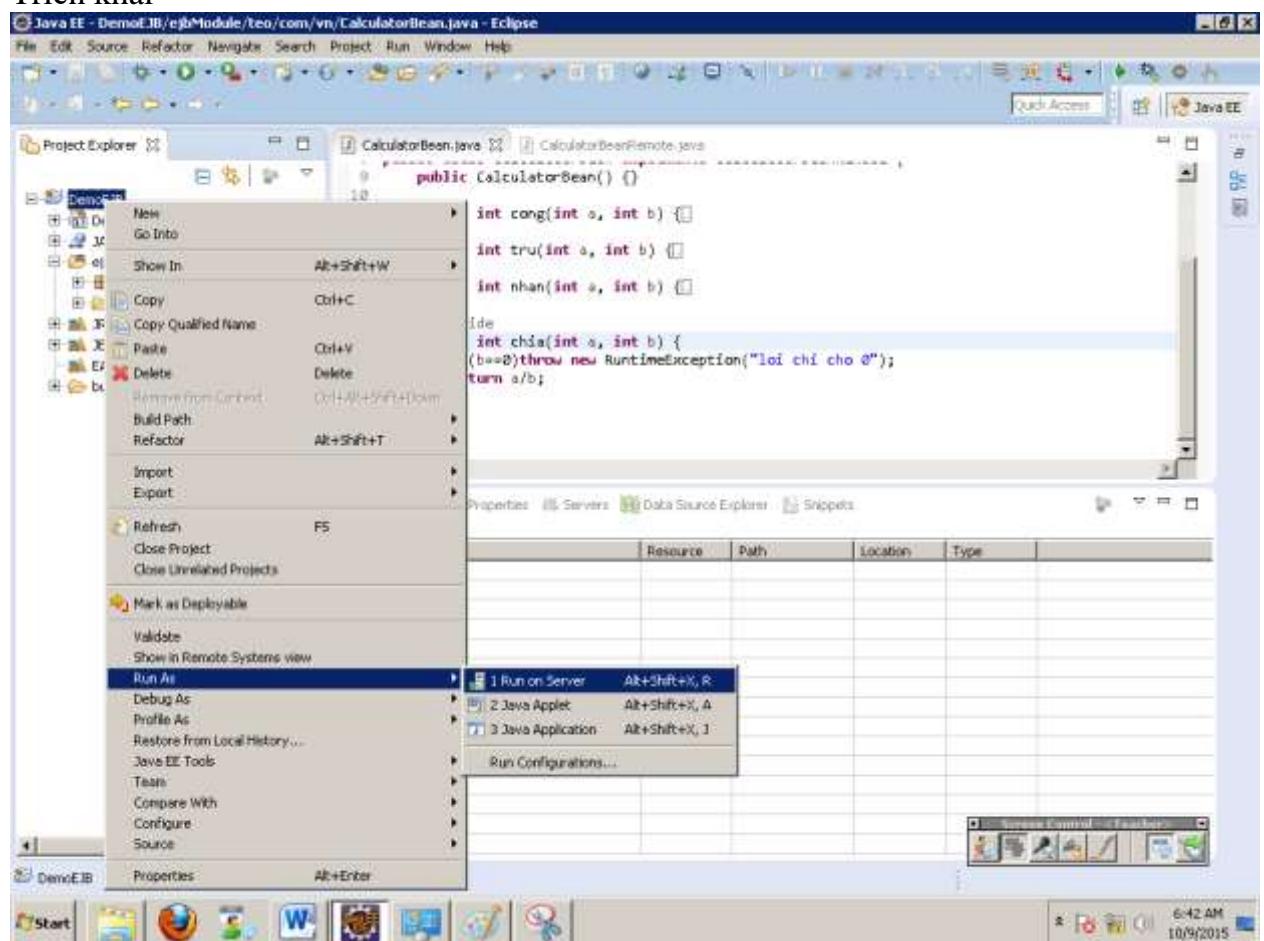
Viết lớp implement

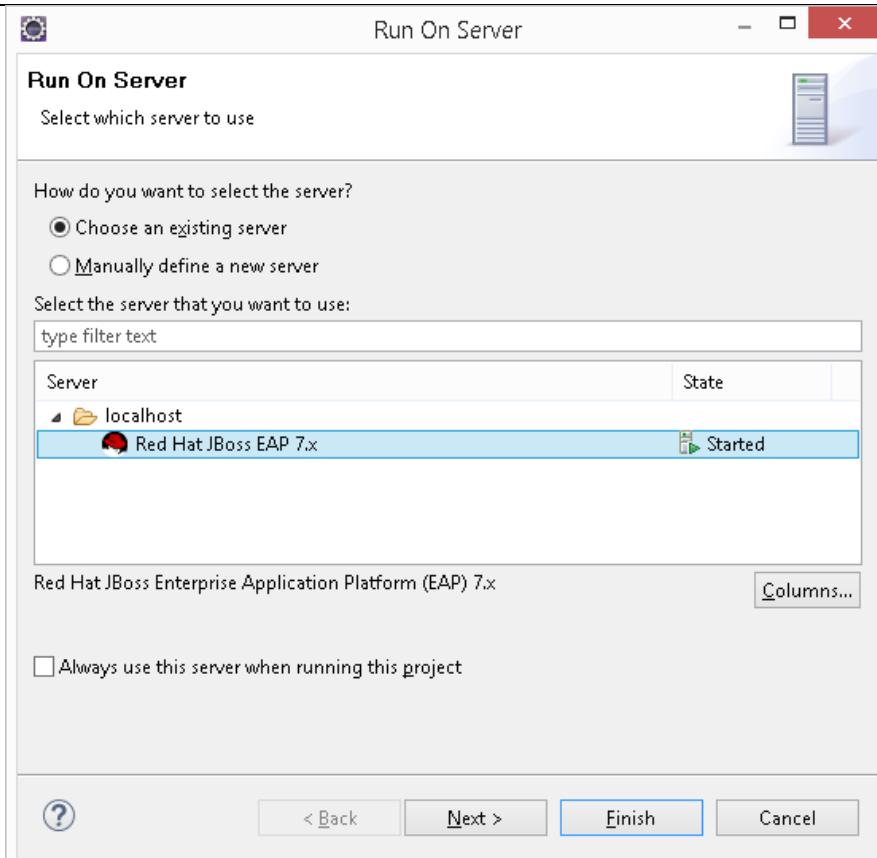
```

CalculatorBean.java ✘ CalculatorBeanRemote.java
1
2
3+ import javax.ejb.LocalBean;[]
4
5
6 @Stateless
7 @LocalBean
8 public class CalculatorBean implements CalculatorBeanRemote {
9     public CalculatorBean() {}
10
11+     public int cong(int a, int b) {}[]
12+
13+     public int tru(int a, int b) {}[]
14+
15+     public int nhan(int a, int b) {}[]
16+
17+     public int chia(int a, int b) {
18         if(b==0)throw new RuntimeException("loi chi cho 0");
19         return a/b;
20     }
21
22+
23+
24+
25+
26+
27+
28+
29+
30+
31+
32+

```

4. Triển khai





Nhấn Finish.

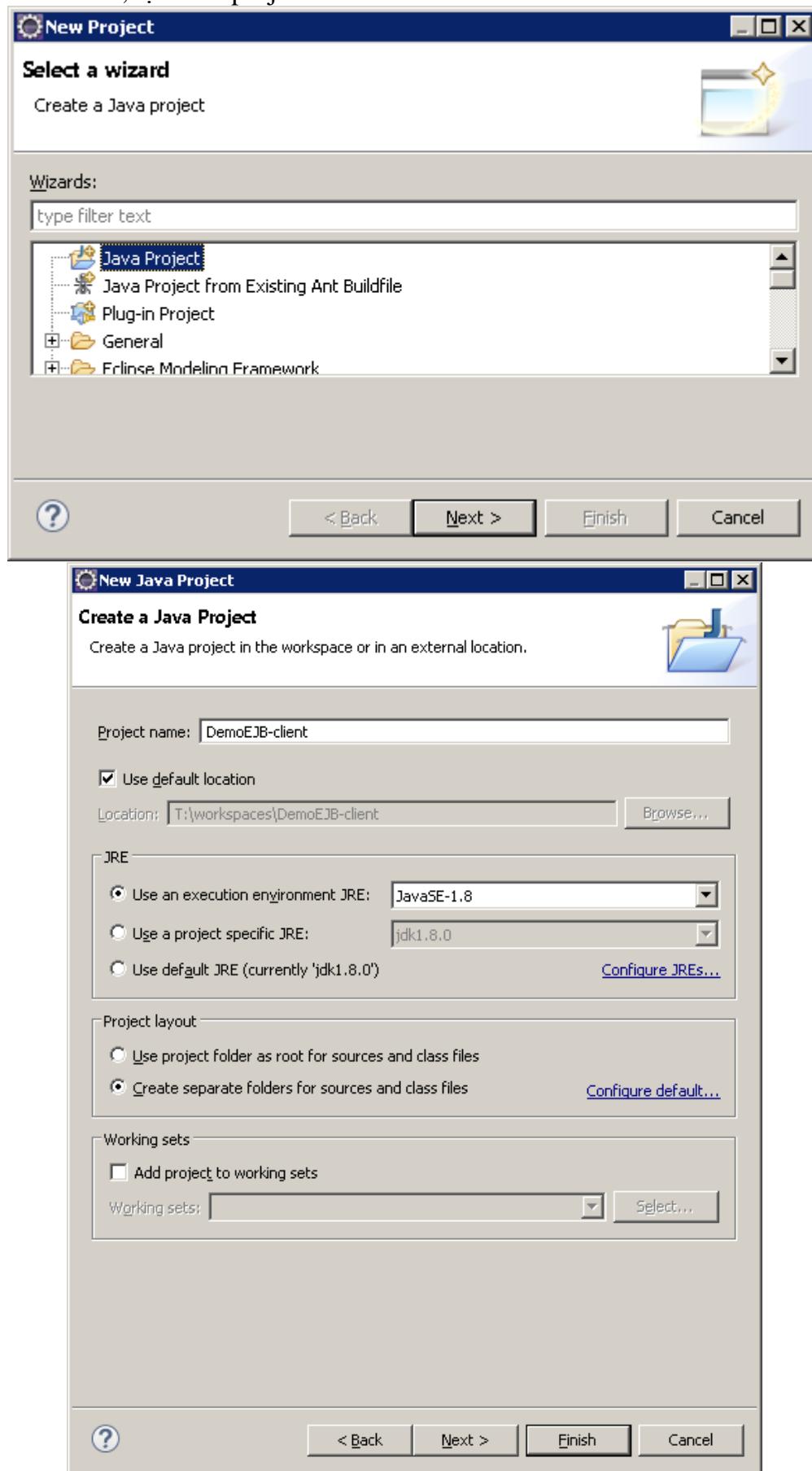
The screenshot shows the Eclipse IDE's Console view. The title bar includes tabs for 'Markers', 'Properties', 'Servers', 'Data Source Explorer', 'Snippets', and 'Console'. The main area displays deployment logs for 'Boss EAP 6.1+ [JBoss Application Server Startup Configuration]'. The log output includes several lines starting with 'java:' followed by class names like 'CalculatorBeanRemote' and 'CalculatorBean'. A red box highlights these lines. The log continues with other deployment messages and starts of the JBoss Web Server.

```

Boss EAP 6.1+ [JBoss Application Server Startup Configuration] T:[Java]\jdk1.8.0\bin\javaw.exe (Oct 9, 2015, 6:43:33 AM)
26:43:45,397 INFO  [org.jboss.as.ejb3.deployment.processors.EjbJndiBindingsDeploymentUnitProcessor]
java:global/DemoEJB/CalculatorBean!teo.com.vn.CalculatorBeanRemote
java:app/DemoEJB/CalculatorBean!teo.com.vn.CalculatorBeanRemote
java:module/CalculatorBean!teo.com.vn.CalculatorBeanRemote
java:jboss/exported/DemoEJB/CalculatorBean!teo.com.vn.CalculatorBeanRemote
java:global/DemoEJB/CalculatorBean!teo.com.vn.CalculatorBean
java:app/DemoEJB/CalculatorBean!teo.com.vn.CalculatorBean
java:module/CalculatorBean!teo.com.vn.CalculatorBean
26:43:47,378 INFO  [org.jboss.ws.common.management] (MSC service thread 1-3) JBWS022052: Starting JBoss Web Server
26:43:47,544 INFO  [org.hornetq.core.server] (ServerService Thread Pool -- 60) HQ221020: Started Net
26:43:47,550 INFO  [org.hornetq.core.server] (ServerService Thread Pool -- 60) HQ221020: Started Net

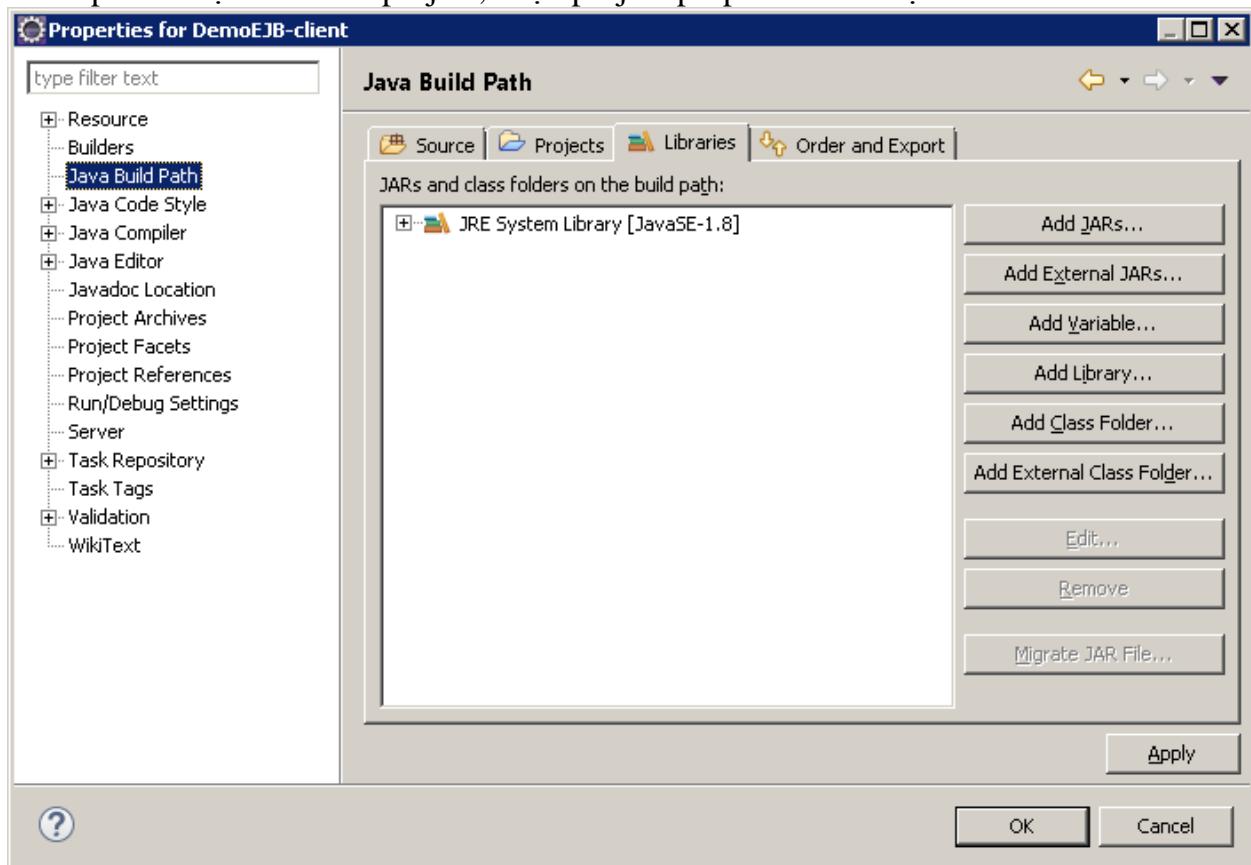
```

1. Chọn Perspective Java, tạo new project

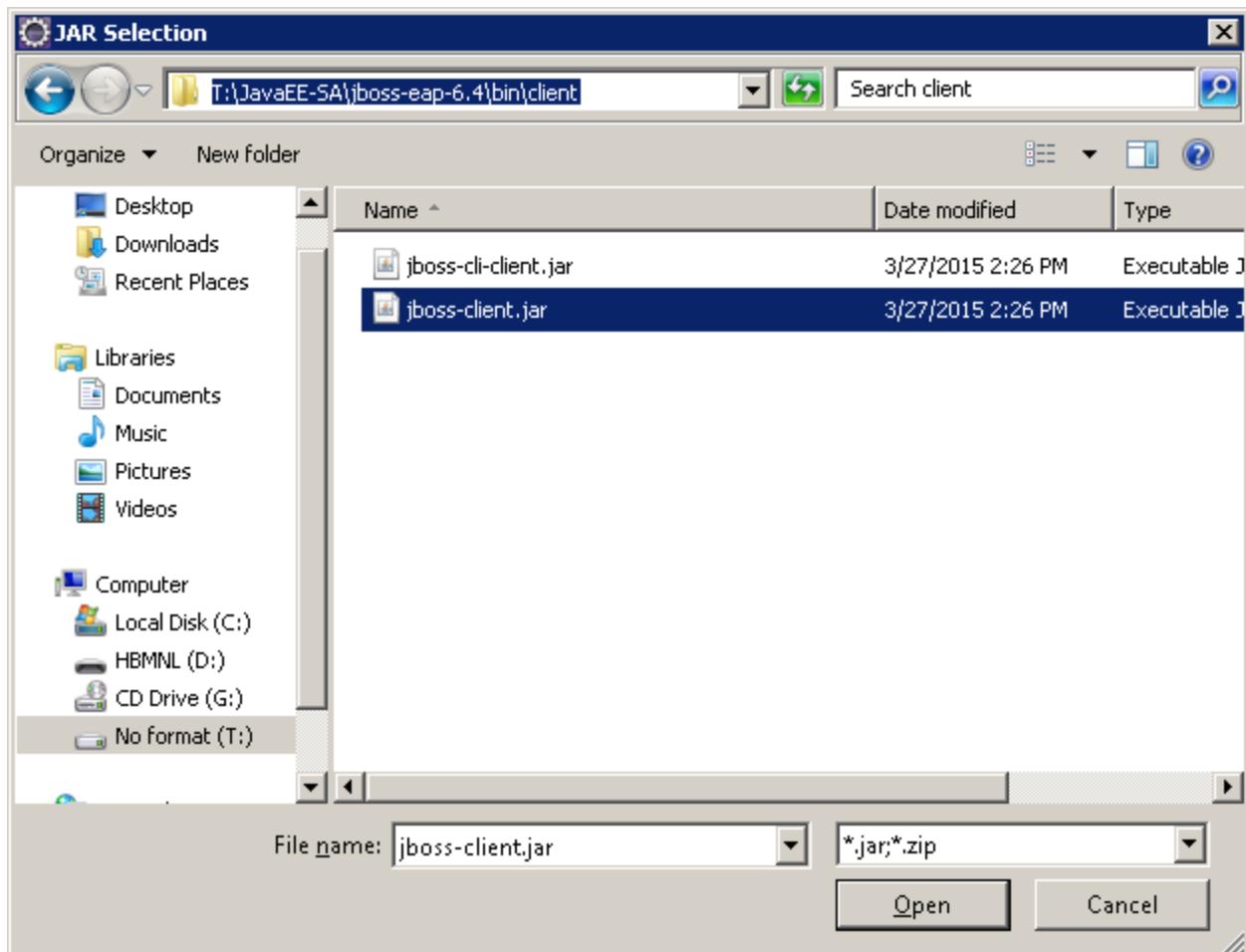


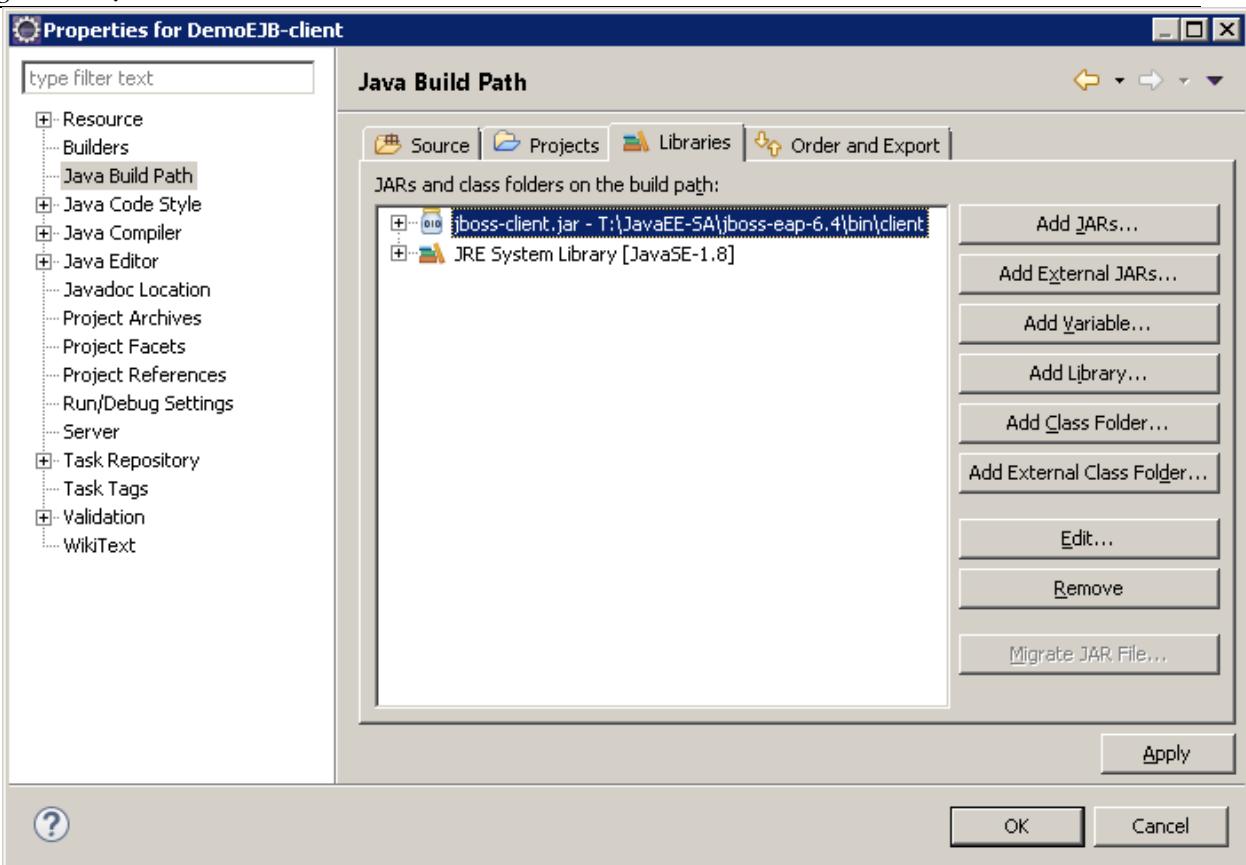
2. Cấu hình thư viện cho client

Nhấn phải chuột lên client project, chọn project properties rồi chọn Java BuildPath

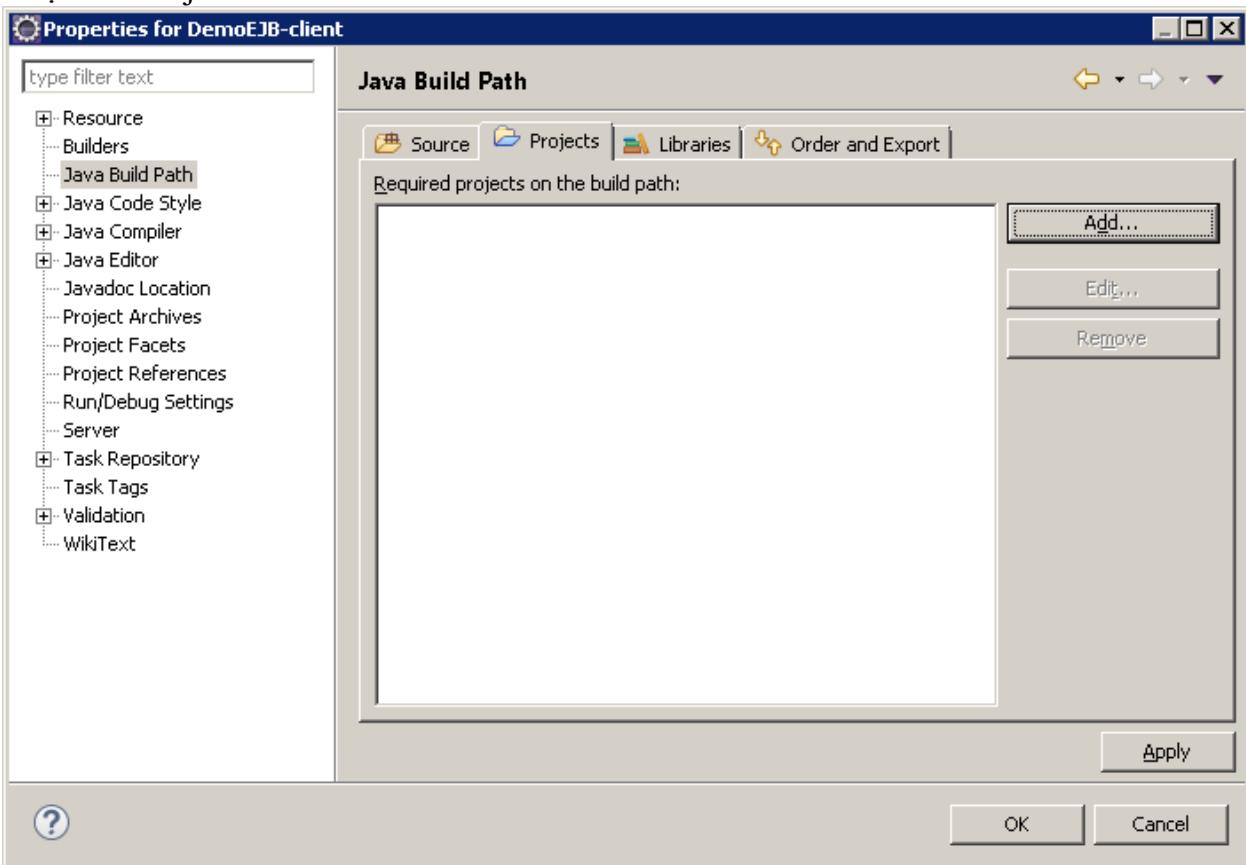


Nhấn nút Add External Jars

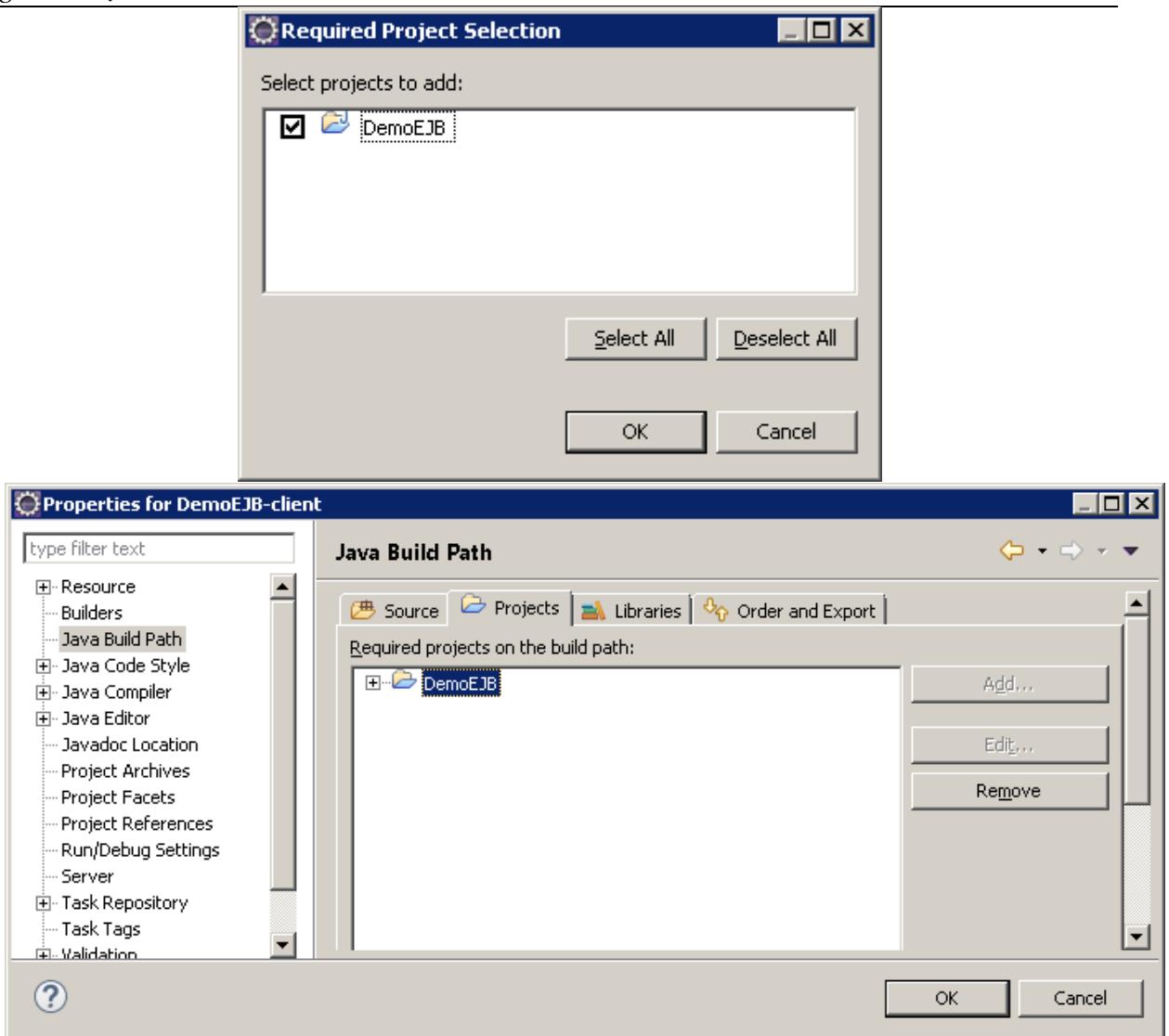




Chọn thẻ Projects



Nhấn nút Add



3. Code cho client

JBoss EAP 6x ta dùng

```

1 package teo.com;
2
3 import java.util.Properties;
4 import javax.naming.Context;
5 import javax.naming.InitialContext;
6 import org.jboss.naming.remote.client.InitialContextFactory;
7 import teo.com.vn.CalculatorBeanRemote;
8
9 public class StatelessSessionDemo {
10     public static void main(String[] args) throws Exception{
11         Properties props=new Properties();
12         props.put(Context.INITIAL_CONTEXT_FACTORY, InitialContextFactory.class.getName());
13         props.put(Context.PROVIDER_URL, "remote://localhost:4447");
14         props.put("jboss.naming.client.ejb.context", true);
15         Context ctx=new InitialContext(props);
16
17         CalculatorBeanRemote calc=(CalculatorBeanRemote)
18             ctx.lookup("DemoEJB/CalculatorBean!teo.com.vn.CalculatorBeanRemote");
19         int kq = calc.chia(6,3);
20         System.out.println(kq);
21     }
22 }
23 }
```

Với JBoss EAP 7.x dùng:

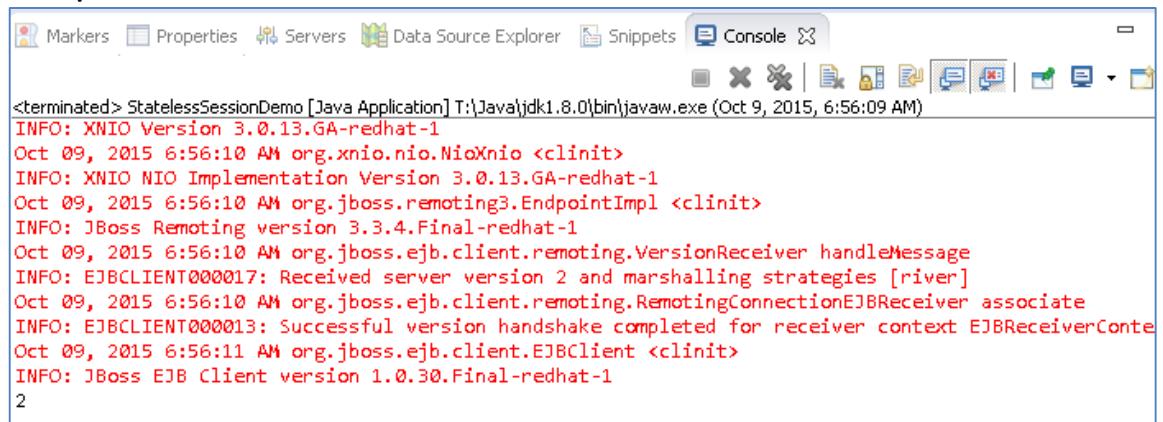
```

public static void main(String[] args) throws Exception{
    Properties prop=new Properties();
    prop.put(Context.INITIAL_CONTEXT_FACTORY,
              "org.jboss.naming.remote.client.InitialContextFactory");
    prop.put(Context.PROVIDER_URL, "http-remoting://localhost:8080");
    prop.put("jboss.naming.client.ejb.context", true);
    Context ctx = new InitialContext(prop);

    Object obj=ctx.lookup("TestEJB/Calc!teo.com.CalcRemote");//ok
    System.out.println(obj);
    CalcRemote calc=(CalcRemote)obj;
    double x=calc.add(3, 4);
    System.out.println(x);
}

```

4. Thực thi client



The screenshot shows the Eclipse IDE's Console view with the following log output:

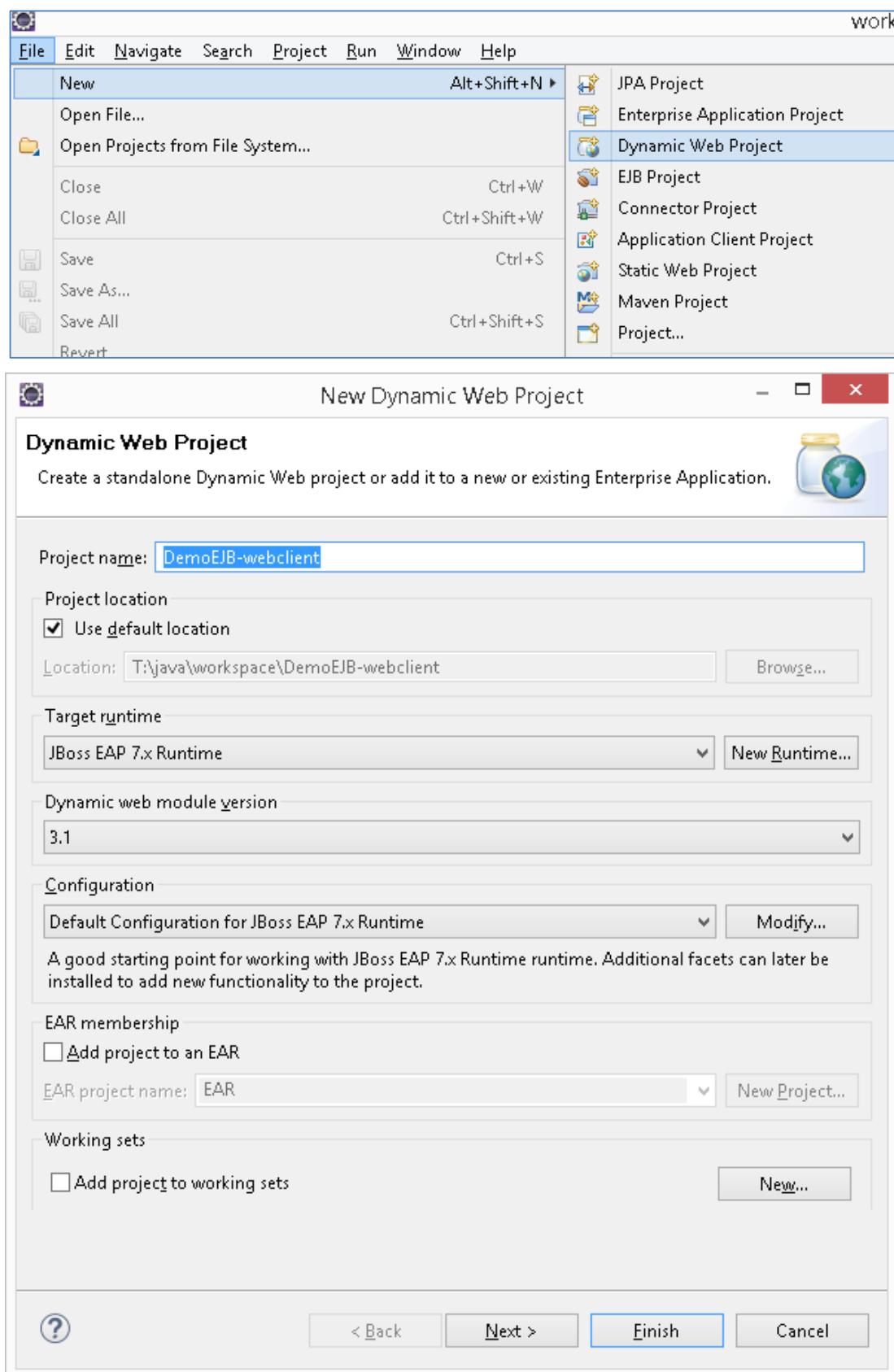
```

<terminated> StatelessSessionDemo [Java Application] T:\Java\jdk1.8.0\bin\javaw.exe (Oct 9, 2015, 6:56:09 AM)
INFO: XNIO Version 3.0.13.GA-redhat-1
Oct 09, 2015 6:56:10 AM org.xnio.nio.NioXnio <clinit>
INFO: XNIO NIO Implementation Version 3.0.13.GA-redhat-1
Oct 09, 2015 6:56:10 AM org.jboss.remoting3.EndpointImpl <clinit>
INFO: JBoss Remoting version 3.3.4.Final-redhat-1
Oct 09, 2015 6:56:10 AM org.jboss.ejb.client.remoting.VersionReceiver handleMessage
INFO: EJBCLIENT000017: Received server version 2 and marshalling strategies [river]
Oct 09, 2015 6:56:10 AM org.jboss.ejb.client.remoting.RemotingConnectionEJBReceiver associate
INFO: EJBCLIENT000013: Successful version handshake completed for receiver context EJBReceiverConte
Oct 09, 2015 6:56:11 AM org.jboss.ejb.client.EJBClient <clinit>
INFO: JBoss EJB Client version 1.0.30.Final-redhat-1
2

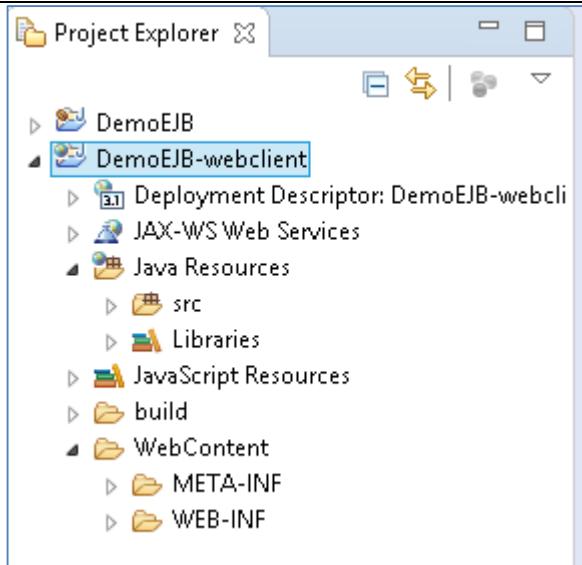
```

WEB CLIENT

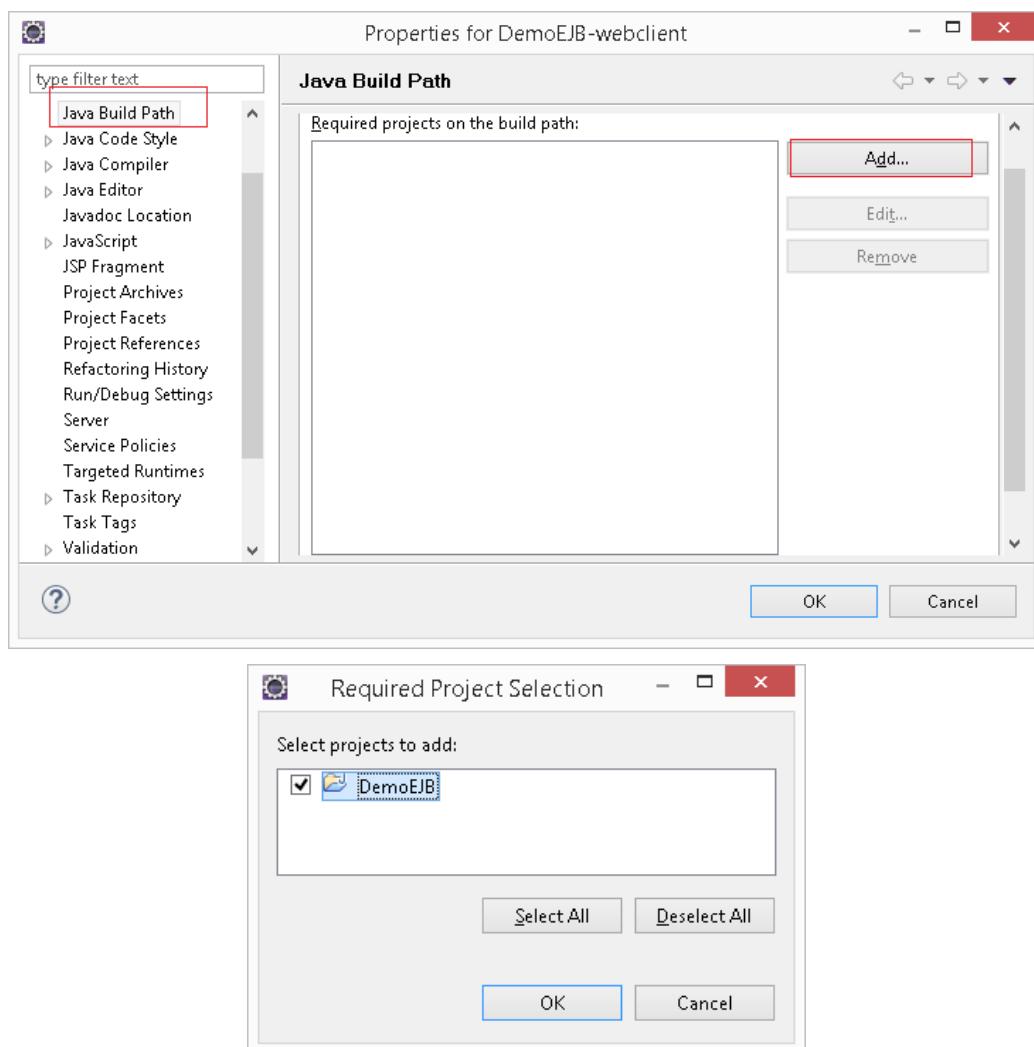
Chọn lại perspective JavaEE sau đó tạo 1 Dynamic Web Project



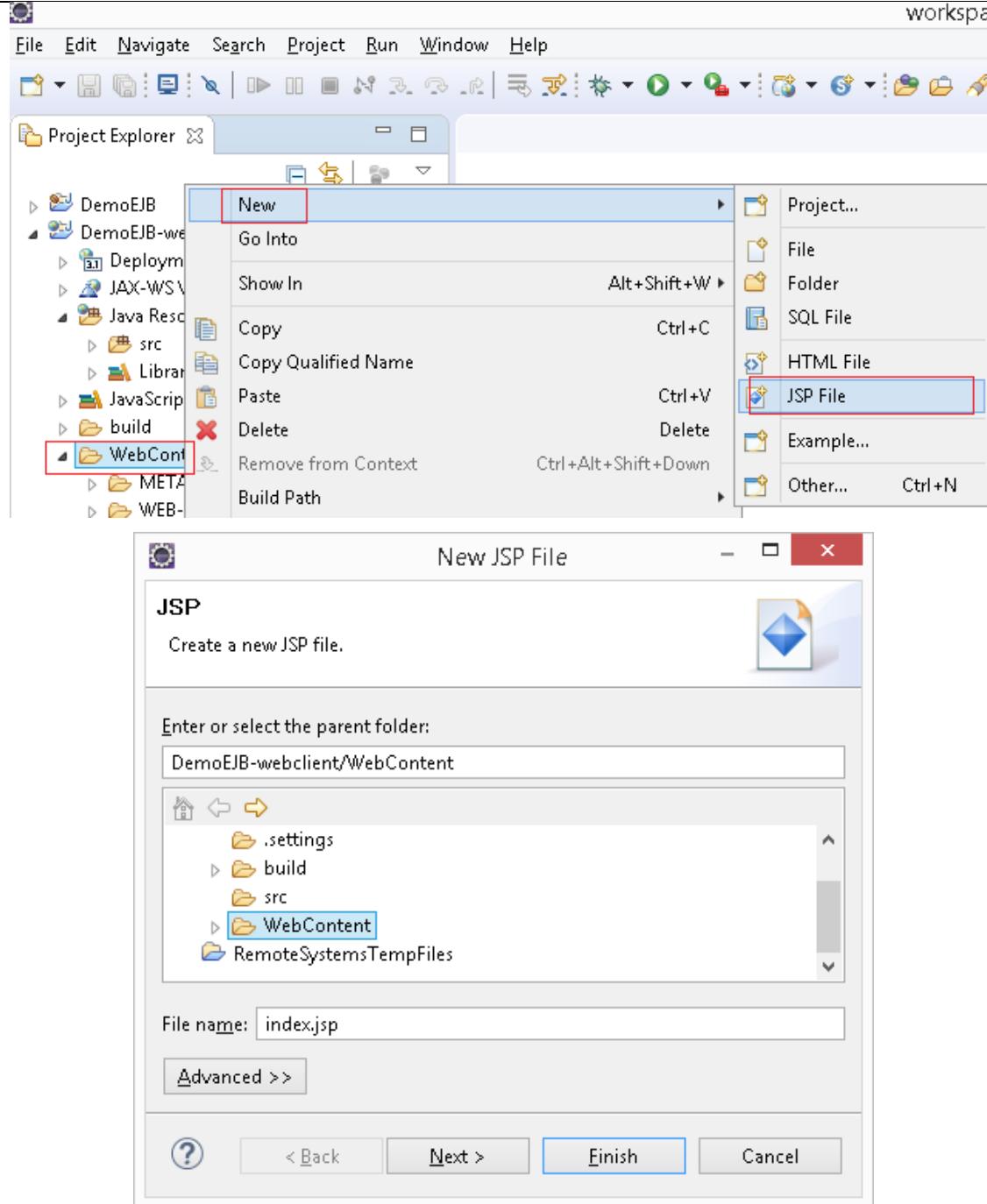
Nhấn Finish ta được



Thêm tham chiếu đến EJB: nhấn phải chuột lên project, chọn Properties



Ta thêm trang JSP hiển thị phía client: Nhấn phải chuột lên WebContent them trang như hình



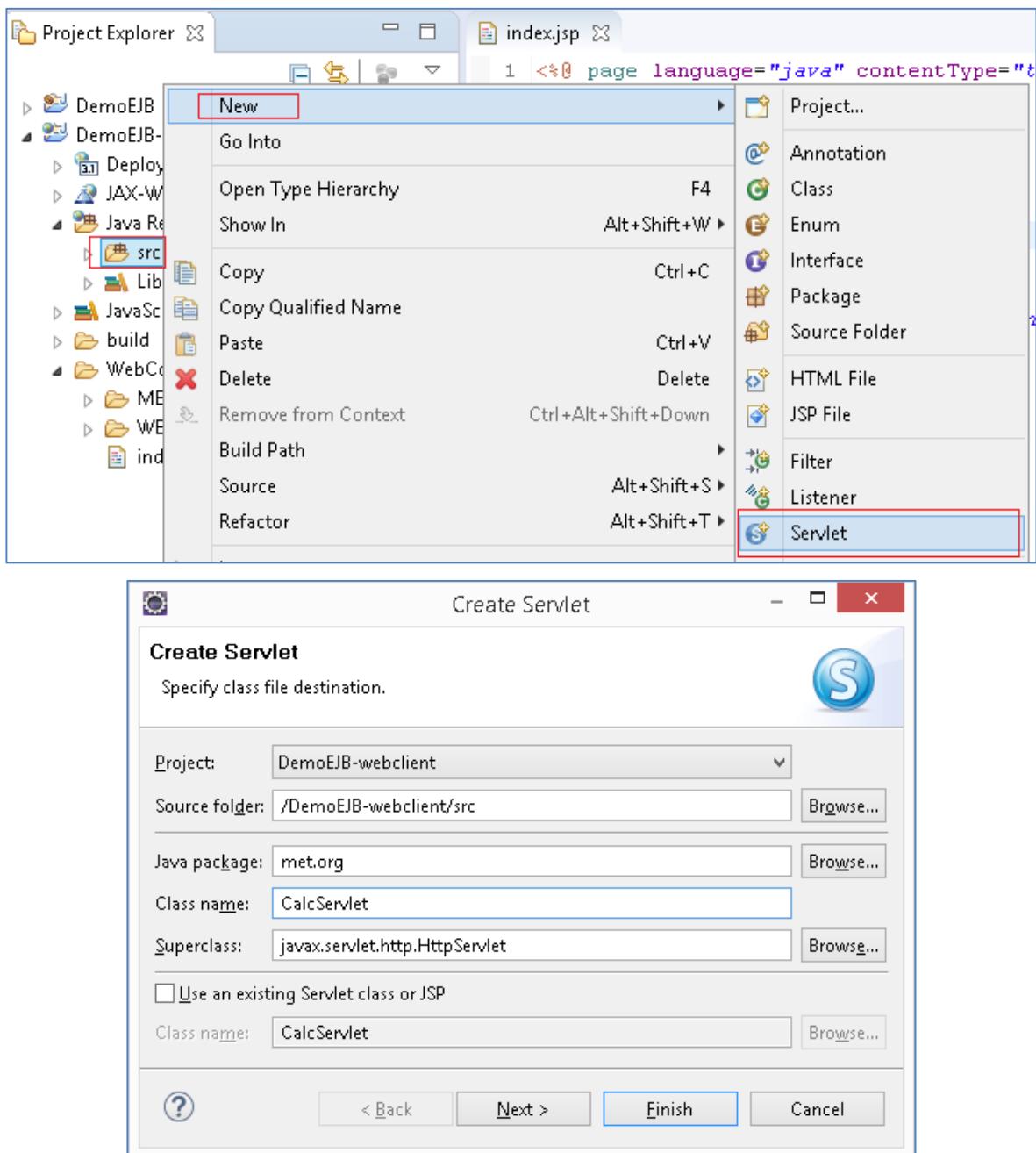
Nội dung trang

```

index.jsp
1 <%@ page language="java" contentType="text/html; charset=UTF-8
2 <html>
3 <head>
4 <title>index</title>
5 </head>
6 <body>
7 <form action="?????> tên của action sẽ được thay bằng servlet
8     Số A:<input name="a"/><br/>
9     Số B:<input name="b"/><br/>
10    <input type="submit" value="Cộng"/>
11 </form>
12 </body>
13 </html>

```

Tạo 1 Servlet xử lý khi người dùng nhấn nút submit: nhấn phải chuột lên thư mục src rồi chọn như hình



```

1 package met.org;
2
3+ import java.io.IOException;
4
5
6 @WebServlet("/CalcServlet")
7 public class CalcServlet extends HttpServlet {
8     private static final long serialVersionUID = 1L;
9
10    public CalcServlet() {
11        super();
12    }
13
14    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws IOException {
15        doPost(request, response);
16    }
17
18 }
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33

```

Chú ý: CalcServlet sẽ là action cho nút submit của trang jsp

Code cho phương thức doGet như sau:

```

1 package met.org;
2
3+ import java.io.IOException;
4
5
6 @WebServlet("/CalcServlet")
7 public class CalcServlet extends HttpServlet {
8     private static final long serialVersionUID = 1L;
9
10    @EJB
11    private CalculatorBeanRemote calc; └── Tham chiếu đến interface của ejb
12
13    public CalcServlet() {
14        super();
15    }
16
17    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws IOException {
18        //Lấy giá trị được nhập trên 2 input-text
19        double a=Double.parseDouble(request.getParameter("a"));
20        double b=Double.parseDouble(request.getParameter("b"));
21
22        //gọi thực thi
23        double kq = calc.cong(a, b);
24        response.getWriter().println(a+" + "+b+" = "+kq);
25    }
26
27
28
29
30    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws IOException {
31    }
32
33 }

```

Sửa code trang jsp, them action cho form như sau

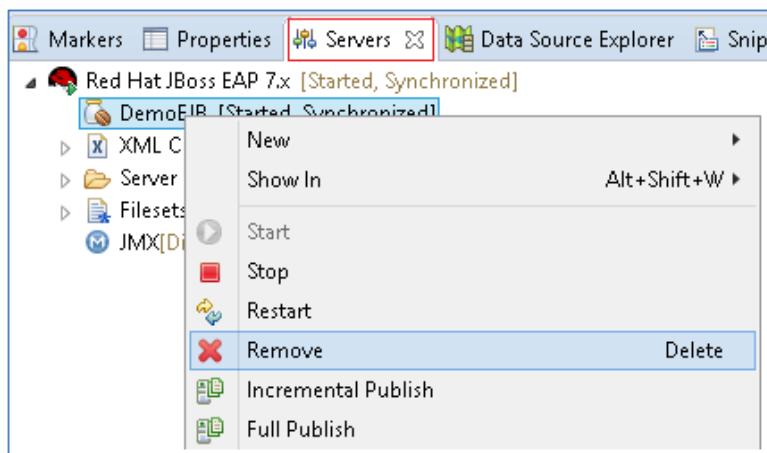
```

1 <%@ page language="java" contentType="text/html; charset=UTF-8" %>
2 <html>
3 <head>
4 <title>index</title>
5 </head>
6 <body>
7 <form action="CalcServlet">
8     Số A:<input name="a"/><br/>
9     Số B:<input name="b"/><br/>
10    <input type="submit" value="Cộng"/>
11 </form>
12 </body>
13 </html>

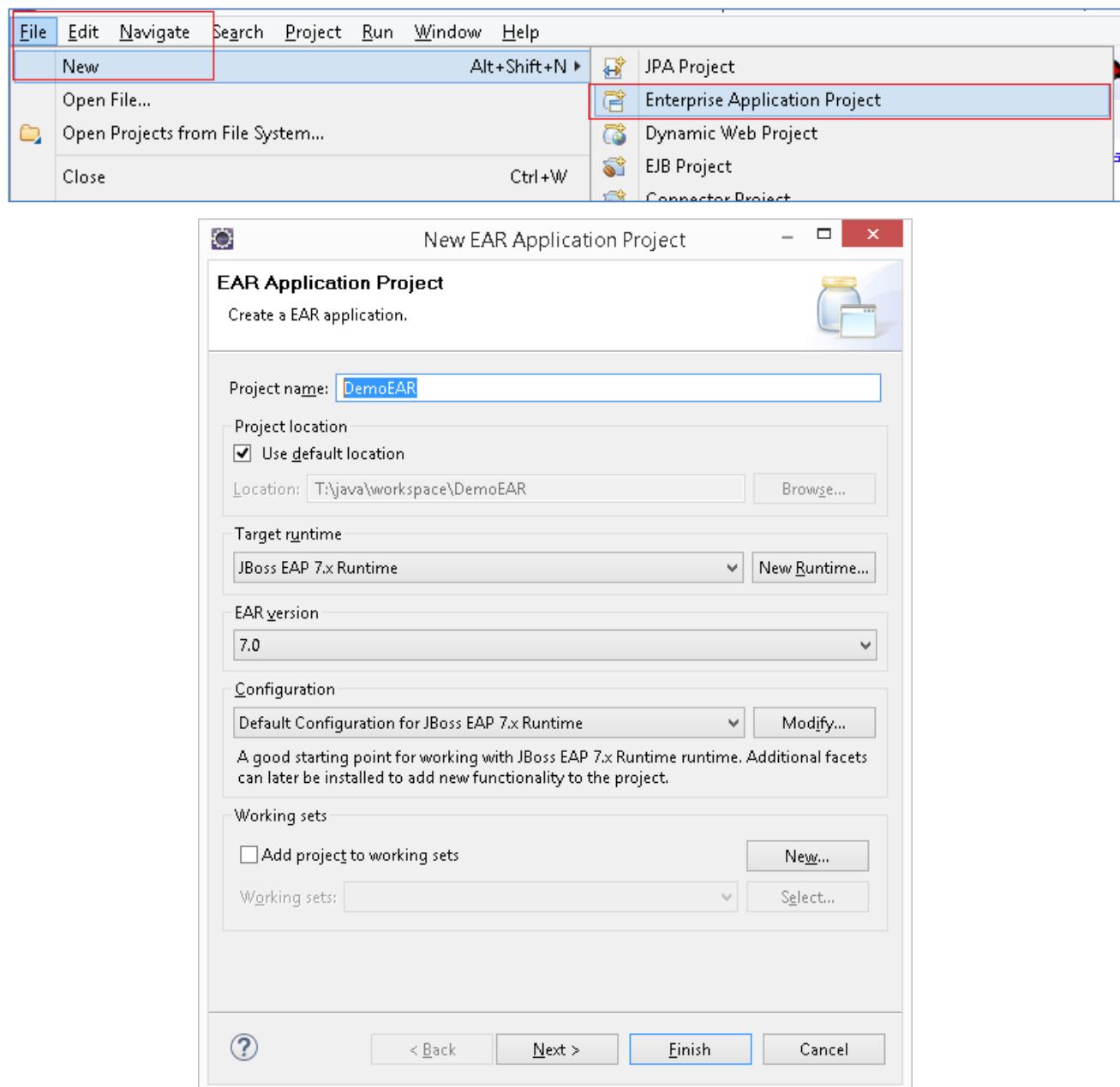
```

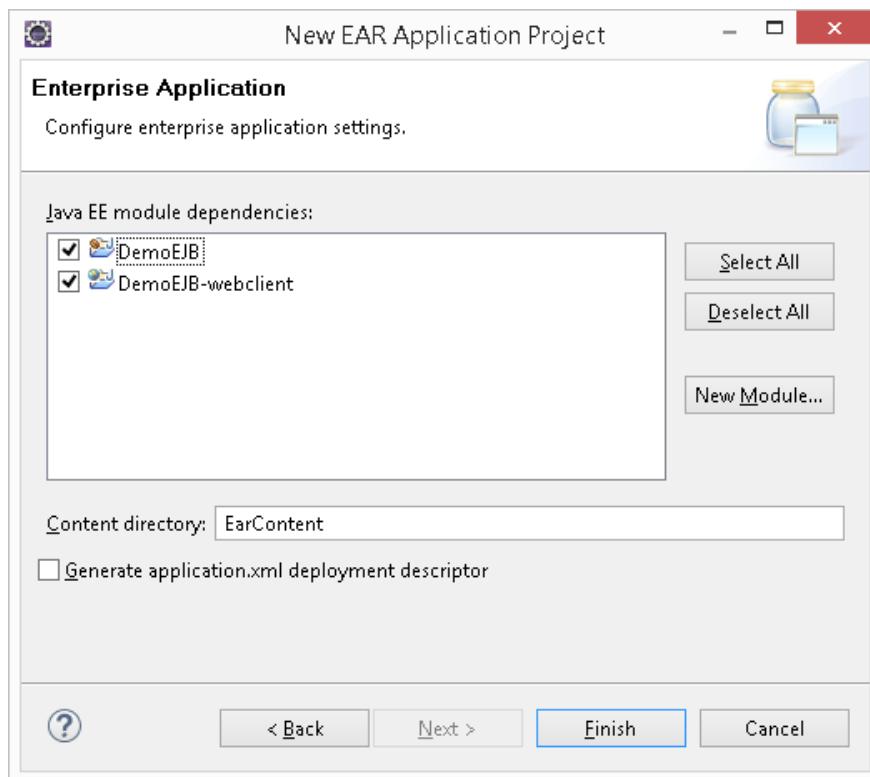

ENTERPRISE APPLICATION PROJECT

Nếu bạn đã triển khai EJB rồi thì nên gỡ bỏ nó đi bằng cách nhấp phải chuột lên project đã triển khai và chọn Remove như hình



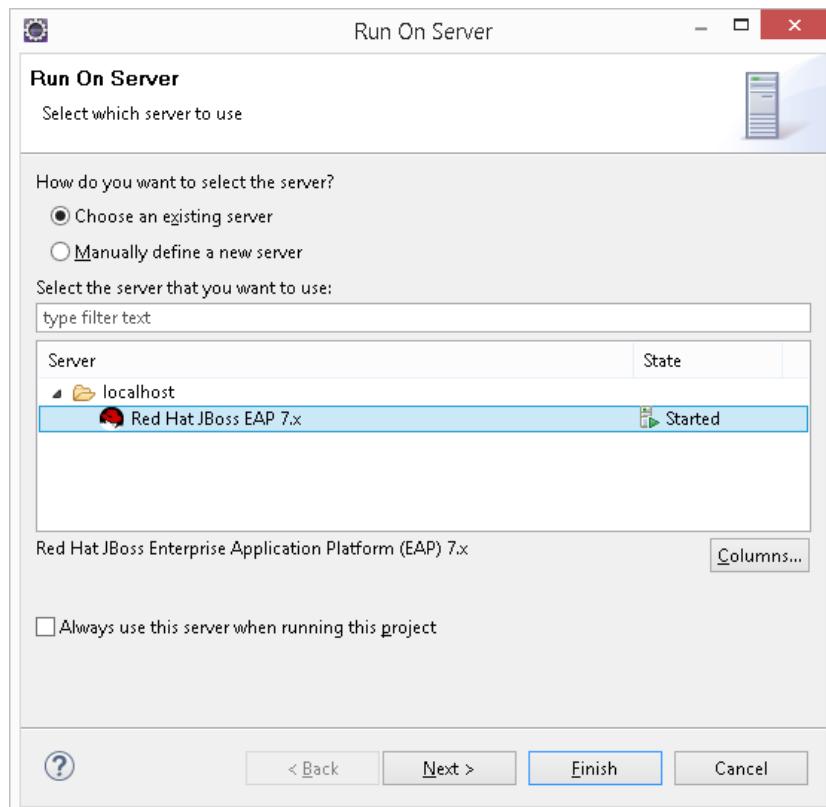
Sau đó tạo mới một Enterprise Application Project





Chọn 2 module như hình sau đó nhấn Finish.

Sau đó triển khai EAProject này bằng cách nhấn phải chuột lên nó, chọn Run As->Run on Server



Nhấn Finish. JBoss sẽ triển khai cả ejb lẫn web client và thực thi trang index.jsp của web client

A screenshot of a web browser window titled "index" with the URL "http://localhost:8080/DemoEJB-webclient/". The page contains two input fields: "Số A:" with the value "6" and "Số B:" with the value "7". Below the inputs is a button labeled "Cộng" (Add). The browser interface includes standard navigation buttons (back, forward, stop, refresh) and a status bar.

Nhấn nút Cộng, kết quả như sau

A screenshot of a web browser window titled "http://localhost:8080/DemoEJB-webclient/CalcServlet?a=6&b=7" with the URL "http://localhost:8080/DemoEJB-webclient/CalcServlet?a=6&b=7" in the address bar. The page displays the result of the addition: "6.0 + 7.0 = 13.0". The browser interface includes standard navigation buttons and a status bar.

BÀI TẬP

1. Tạo một stateless session cho việc chuyển đổi tiền tệ từ 1 loại tiền tệ này thành một loại khác. (giữa USD/VND/EUR). Viết chương trình client để thử.
- 2.

III. STATEFULL SESSION BEAN

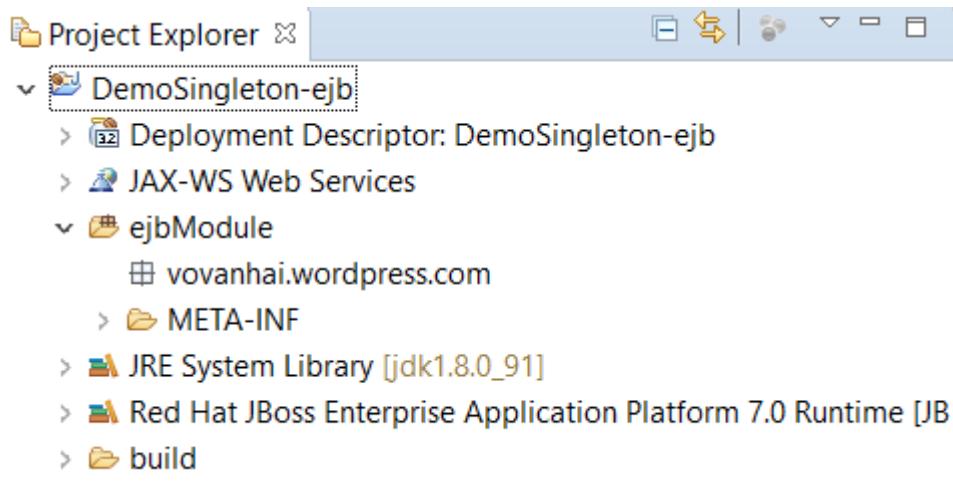
Ví dụ này làm một giỏ hàng

IV. SINGLETON SESSION BEAN

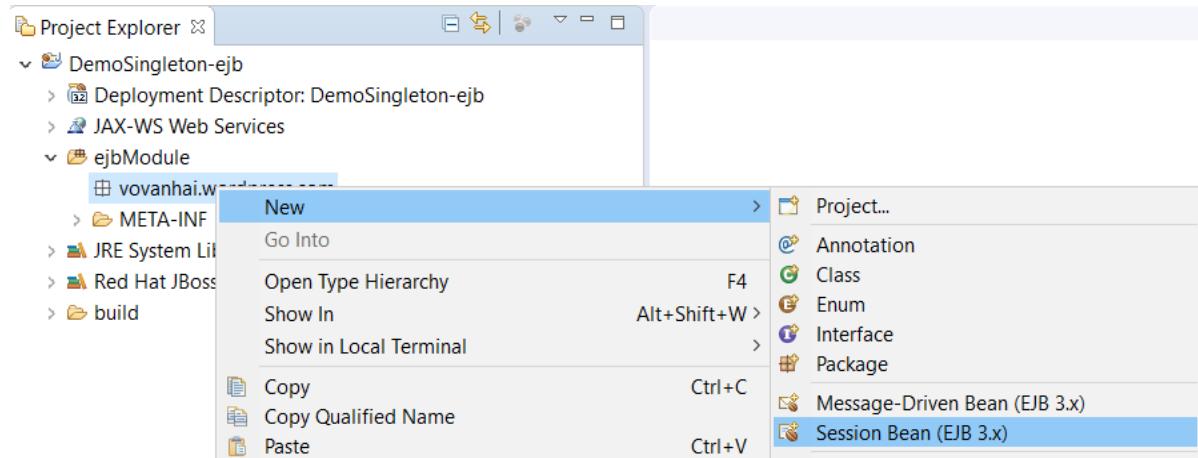
Singleton session bean là một loại bean mà chỉ có duy nhất một instance chia sẻ chung trong vòng đời của ejb

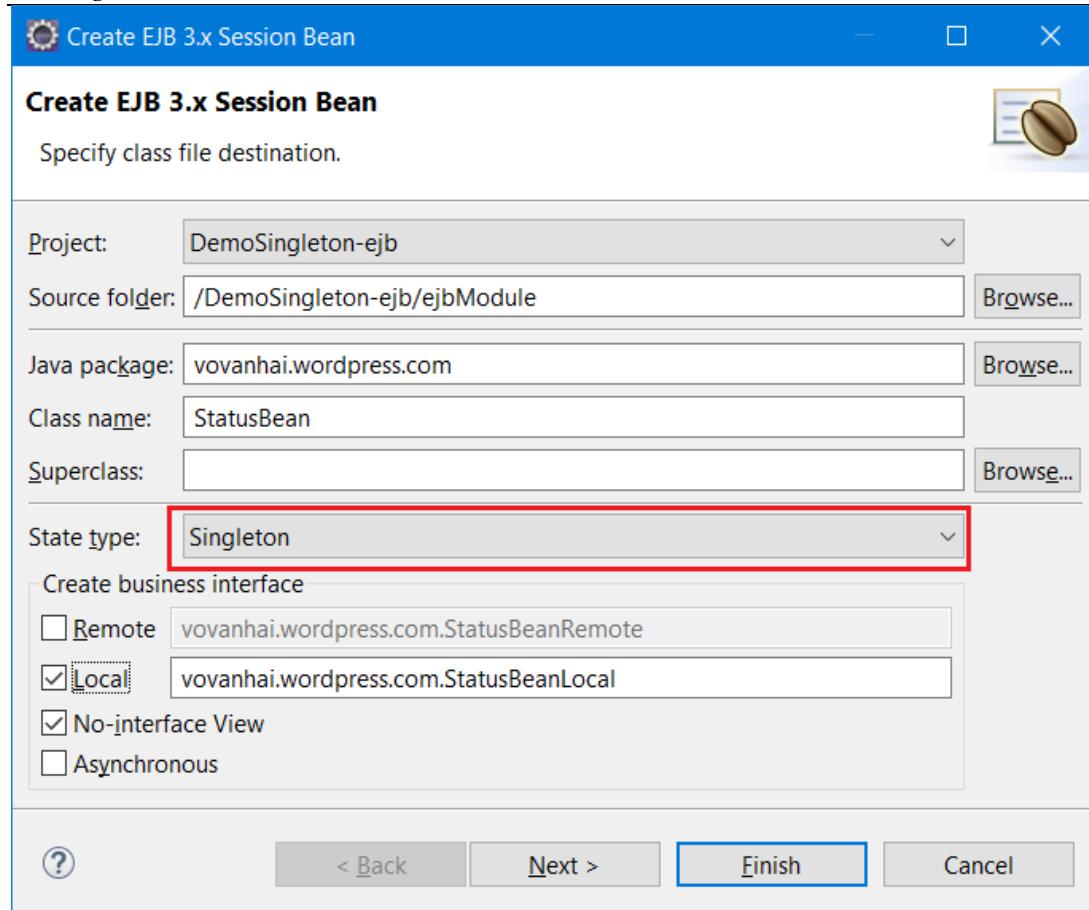
EJB PROJECT

Tạo 1 EJB Project có tên DemoSingleton-ejb như hình

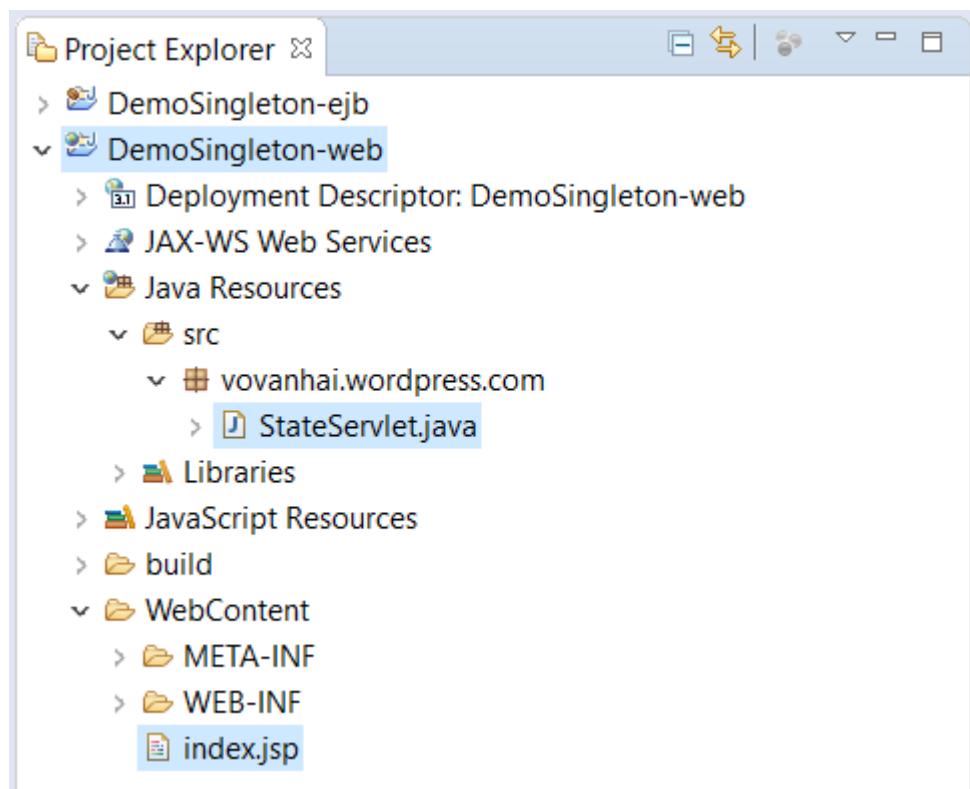


Nhấn chuột phải lên ejbModule vào tạo 1 SessionBean





Tạo 1 Dynamic Web Project có tên DemoSingleton-web sau đó tạo trang index.jsp vào một servlet có tên StateServlet như hình:



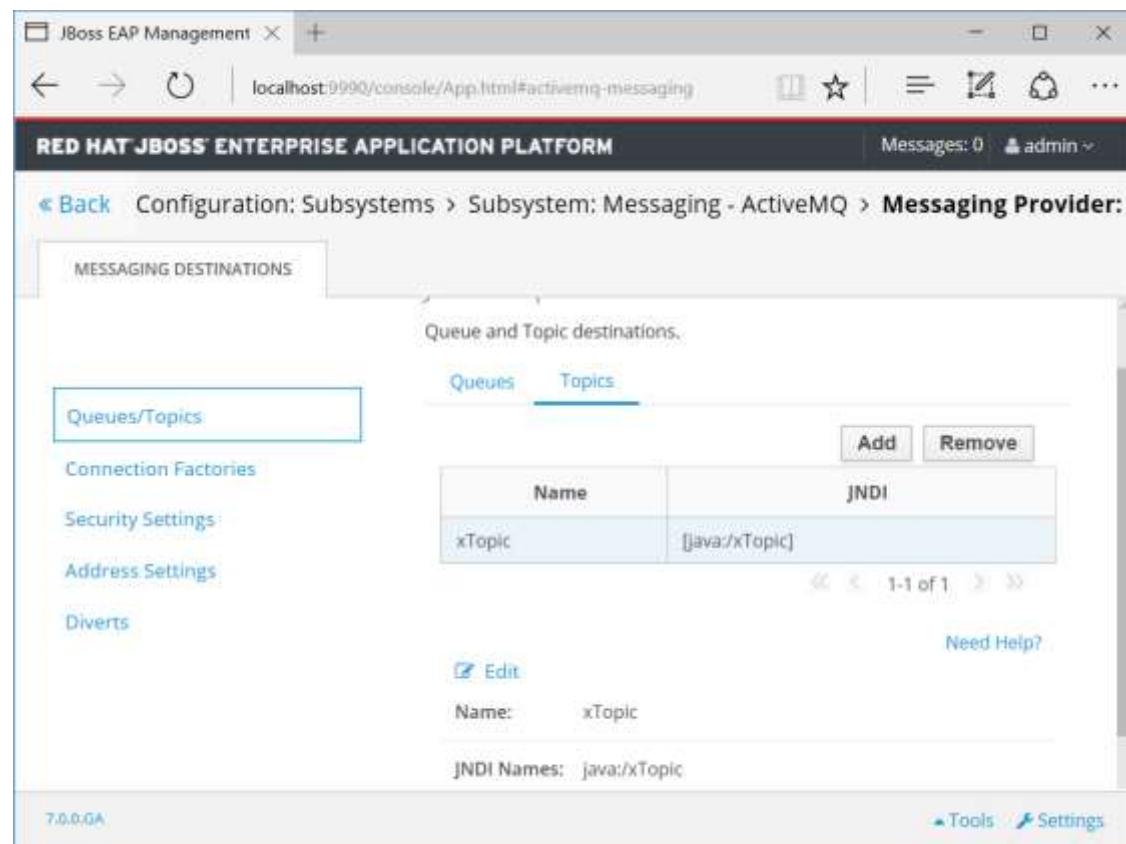
Trang index.jsp có nội dung

V. MESSAGE-DRIVEN-BEAN

CẤU HÌNH

Trong %JBOSS_HOME%\standalone\configuration, mở file standalone-full.xml, tìm đến phần đoạn `<subsystem xmlns="urn:jboss:domain:messaging-activemq:1.0">` rồi thêm 2 dòng như hình dưới để tạo 1 queue tên xQueue và một topic tên xTopic

Bạn cũng có thể dùng Admin Console để thêm hoặc cấu hình



Cho qui trình xử lý đơn hàng được mô tả như sau:

Hệ thống dùng một Message-Driven Bean lắng nghe client đặt hàng trên queue có tên orderQueue. Khi một client gửi đến một đơn hàng, MDB này sẽ nhận và gọi đến một stateless session bean để xử lý đơn hàng.

Tại stateless session bean này sẽ dùng JPA để lưu đơn hàng xuống csdl và gửi về một đặc tả đơn hàng dạng XML cho MDB. Sau đó MDB sẽ gửi file XML này về cho client.

Phía client cho phép người dùng chọn các sản phẩm, số lượng sản phẩm đưa vào giỏ hàng và sau đó đặt hàng.

Hãy hiện thực một ứng dụng phía server dùng công nghệ EJB

Hướng dẫn: Cho các lớp entities như sau

```
package entities;

import java.io.Serializable;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.FetchType;
import javax.persistence.Id;
import javax.persistence.ManyToOne;
import javax.persistence.Table;
import javax.xml.bind.annotation.XmlRootElement;
import javax.xml.bind.annotation.XmlType;

@XmlRootElement
@XmlType(propOrder={"productID", "quantity", "price", "note"})
@Entity @Table(name="OrderItem")
public class OrderItem implements Serializable{
    @Id
    private long productID; //mã sản phẩm

    @Column(name="quantity", nullable=false)
    private int quantity; //số lượng

    @Column(name="price", nullable=false)
    private double price; //giá bán thực tế

    @Column(name="note", length=200)
    private String note; //ghi chú nếu có

    @ManyToOne(fetch=FetchType.EAGER)
    private Order order;

    public OrderItem() {
    }
    public OrderItem(long productID, int quantity, double price, String
note) {
        this.productID = productID;
```

```
        this.quantity = quantity;
        this.price = price;
        this.note = note;
    }
    public long getProductID() {
        return productID;
    }
    public void setProductID(long productID) {
        this.productID = productID;
    }
    public int getQuantity() {
        return quantity;
    }
    public void setQuantity(int quantity) {
        this.quantity = quantity;
    }
    public double getPrice() {
        return price;
    }
    public void setPrice(double price) {
        this.price = price;
    }
    public String getNote() {
        return note;
    }
    public void setNote(String note) {
        this.note = note;
    }
    @Override
    public int hashCode() {
        final int prime = 31;
        int result = 1;
        result = prime * result + (int) (productID ^ (productID >>>
32));
        return result;
    }
    @Override
    public boolean equals(Object obj) {
        if (this == obj)
            return true;
        if (obj == null)
            return false;
        if (getClass() != obj.getClass())
            return false;
        OrderItem other = (OrderItem) obj;
        if (productID != other.productID)
            return false;
        return true;
    }
    @Override
    public String toString() {
```

```

        return "OrderItem [productID=" + productID + ", quantity=" +
quantity + ", price=" + price + ", note=" + note
                + "]";
    }
}

package entities;

import java.io.Serializable;
import java.util.Date;
import java.util.ArrayList;
import javax.persistence.CascadeType;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.FetchType;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.OneToMany;
import javax.persistence.Table;
import javax.xml.bind.annotation.XmlRootElement;
import javax.xml.bind.annotation.XmlType;

@XmlRootElement
@XmlType(propOrder={"orderID","customerID","orderDate","staffID","lst
OrderItems"})
@Entity @Table(name="Order")
public class Order implements Serializable{
    @Id @GeneratedValue(strategy=GenerationType.IDENTITY)
    @Column(name="orderID")
    private long orderID;//mã số đơn hàng

    @Column(name="customerID",nullable=false,length=15)
    private String customerID;//mã khách hàng

    @Column(name="orderDate")
    private Date orderDate;//ngày đặt hàng

    @Column(name="staffID")
    private String staffID;//nhân viên lập phiếu

    @OneToMany(mappedBy="order",cascade={CascadeType.ALL},fetch=FetchTy
pe.EAGER)
    private ArrayList<OrderItem>lstOrderItems;//danh sách các mục hàng

    public Order() {
        this(0,"",new Date(0),"");
    }
    public Order(long orderID, String customerID,Date orderDate, String
staffID) {
        this.orderID = orderID;
        this.customerID = customerID;
        this.orderDate = orderDate;
    }
}

```

```
        this.staffID = staffID;
        lstOrderItems=new ArrayList<>();
    }
    public long getOrderID() {
        return orderID;
    }
    public void setOrderID(long orderID) {
        this.orderID = orderID;
    }
    public String getCustomerID() {
        return customerID;
    }
    public void setCustomerID(String customerID) {
        this.customerID = customerID;
    }

    public ArrayList<OrderItem> getLstOrderItems() {
        return lstOrderItems;
    }
    public void setLstOrderItems(ArrayList<OrderItem> lstOrderItems) {
        this.lstOrderItems = lstOrderItems;
    }
    public Date getOrderDate() {
        return orderDate;
    }
    public void setOrderDate(Date orderDate) {
        this.orderDate = orderDate;
    }
    public String getStaffID() {
        return staffID;
    }
    public void setStaffID(String staffID) {
        this.staffID = staffID;
    }
    @Override
    public int hashCode() {
        final int prime = 31;
        int result = 1;
        result = prime * result + (int) (orderID ^ (orderID >>> 32));
        return result;
    }
    @Override
    public boolean equals(Object obj) {
        if (this == obj)
            return true;
        if (obj == null)
            return false;
        if (getClass() != obj.getClass())
            return false;
        Order other = (Order) obj;
        if (orderID != other.orderID)
            return false;
    }
}
```

```

        return true;
    }
    @Override
    public String toString() {
        return "Order [orderID=" + orderID + ", customerID=" +
customerID +
                ", lst=" + lstOrderItems + ", orderDate=" + orderDate
                + ", staffID=" + staffID + "]";
    }
    /**
     * Thêm một mục hàng vào đơn đặt hàng
     * @param item là mục cần thêm. Nếu mục này đã có trong đơn hàng
     * thì cập nhật số lượng. nếu không thì thêm mới
     */
    public void addOrderItem(OrderItem item){
        if(lstOrderItems.contains(item)){
            OrderItem
            oi=lstOrderItems.get(lstOrderItems.indexOf(item));//lấy item đã có
            //cập nhật số lượng
            oi.setQuantity(oi.getQuantity()+item.getQuantity());
        }
        else{
            lstOrderItems.add(item); //chưa có thì thêm mới
        }
    }
    /**
     * Bỏ? một mục hàng kh?i đơn hàng
     * @param item là mục hàng cần b?
     * @return false nếu mục hàng không tồn tại
     */
    public boolean removeItem(OrderItem item){
        if(lstOrderItems.contains(item))
            return lstOrderItems.remove(item);
        return false;
    }
}

```

File persistent.xml có cấu trúc

```

<?xml version="1.0" encoding="UTF-8"?>
<persistence
    xmlns="http://xmlns.jcp.org/xml/ns/persistence" version="2.1"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/persistence
    http://xmlns.jcp.org/xml/ns/persistence/persistence_2_1.xsd">
    <persistence-unit name="your_PU_name" transaction-type="JTA">
        <provider>org.hibernate.ejb.HibernatePersistence</provider>
        <jta-data-source>your_datasource_name</jta-data-source>
        <properties>
            <property
                name="hibernate.dialect"
                value="org.hibernate.dialect.SQLServerDialect"/>
            <property name="hibernate.hbm2ddl.auto" value="create"/>
            <property name="hibernate.show_sql" value="true"/>

```

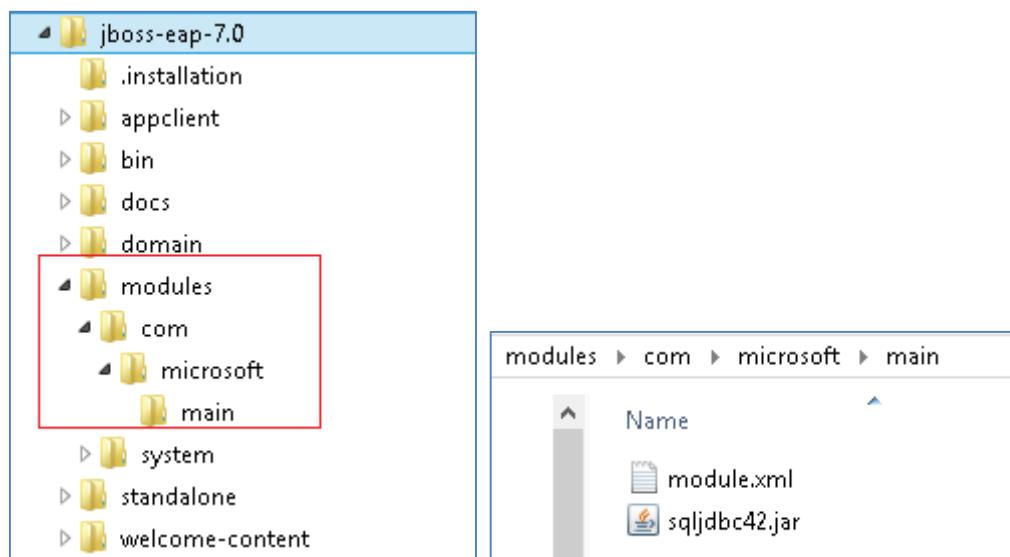
```
</properties>
</persistence-unit>
</persistence>
<!-- hibernate.hbm2ddl.auto =validate | update | create | create-drop --
>
```

VI. JAVA PERSISTENCE API

CẤU HÌNH DATASOURCE TRONG JBossEAP

Cài đặt Microsoft SQL Server JDBC module cho JbossEAP

Tạo cấu trúc thư mục như hình



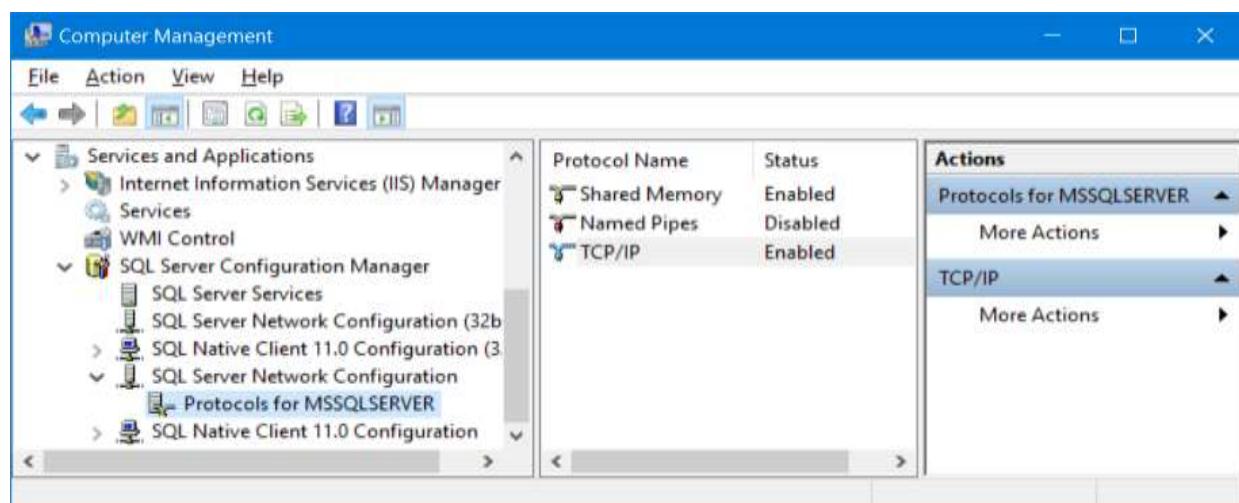
File module.xml có nội dung

```
<module xmlns="urn:jboss:module:1.1" name="com.microsoft">
    <resources>
        <resource-root path="sqljdbc42.jar"/>
    </resources>
    <dependencies>
        <module name="javax.api"/>
        <module name="javax.transaction.api"/>
    </dependencies>
</module>
```

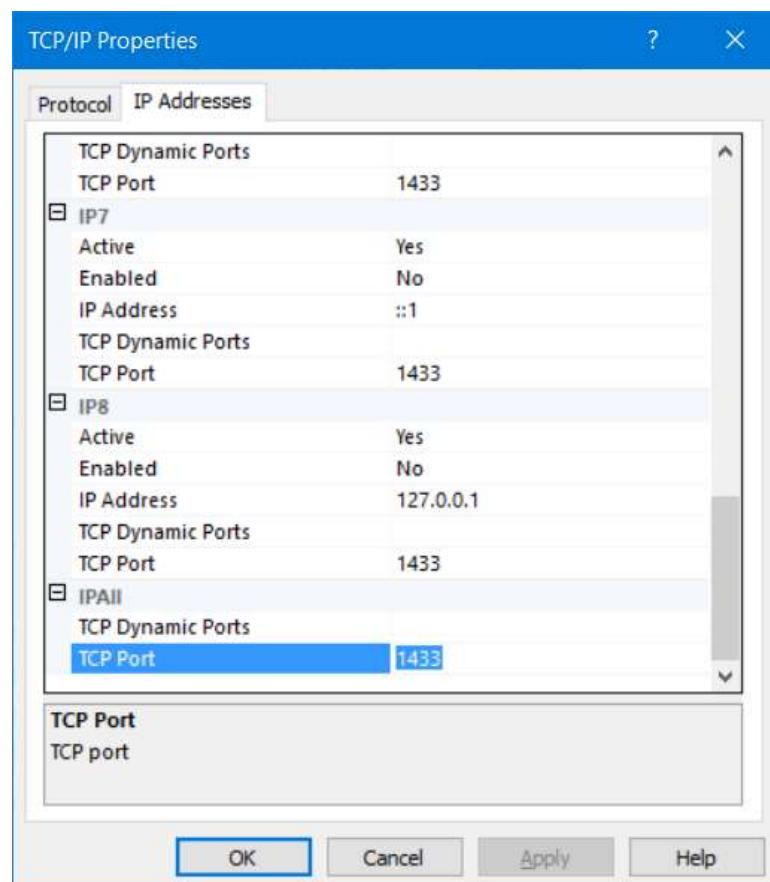
File sqljdbc42.jar là sqlserver driver library

Lưu ý:

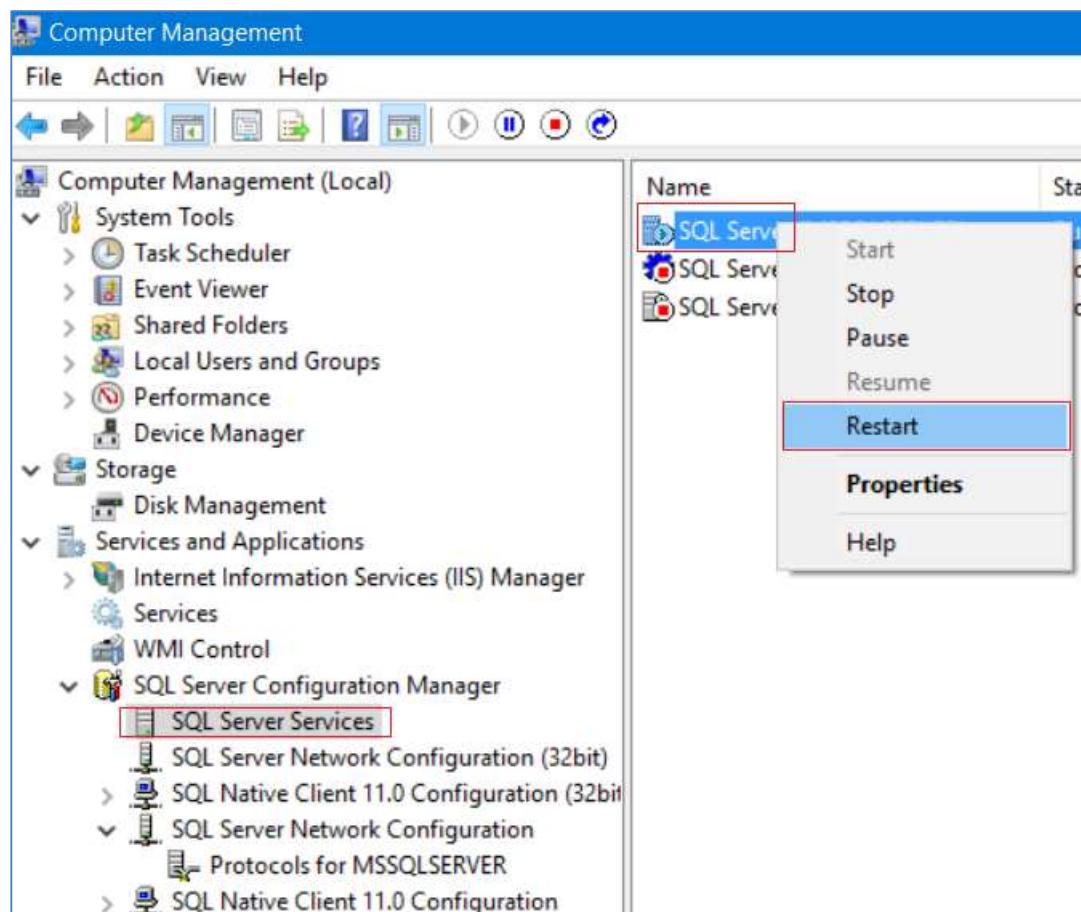
Với MS SQL Server, bạn phải cấu hình cho nó lắng nghe TCP/IP trên cổng 1433 bằng cách: Mở Computer Management → Chọn mục Services and Applications → Chọn SQLServer Configuration manager → SQL Server Network Configuration → Protocols for MSSQLServer như hình.



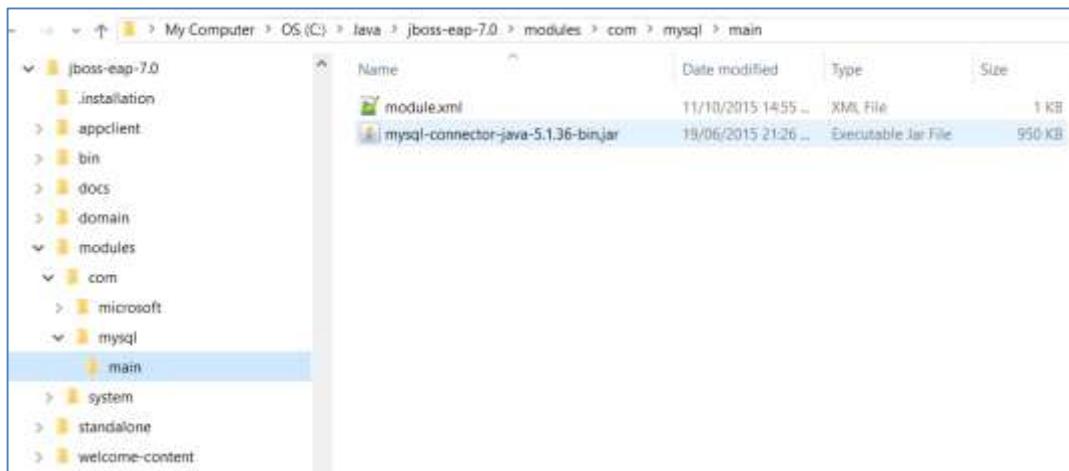
Double click lên TCP/IP để thay đổi properties



Thay TCP Port thành 1433 rồi nhấn OK. Sau đó bạn cần phải restart lại SQL Server service



Cài đặt MySQL JDBC module cho JbossEAP

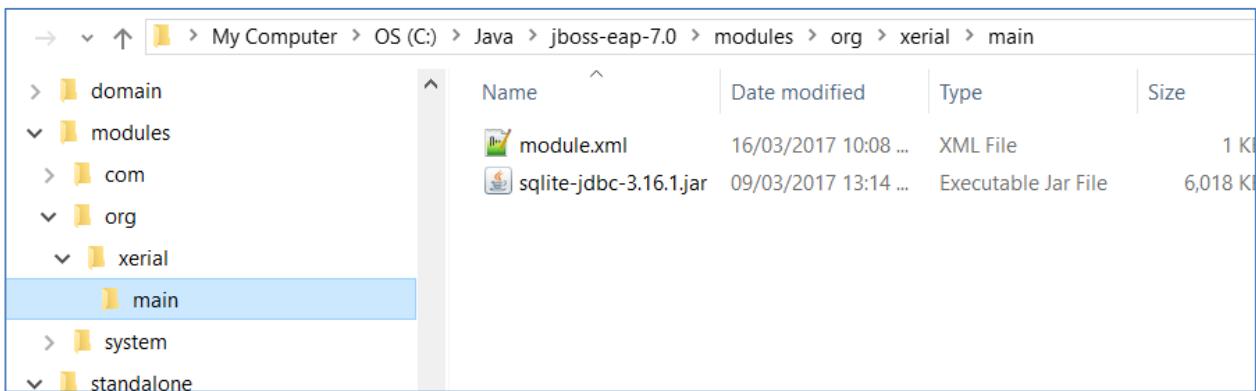


File module.xml có nội dung

```
<?xml version="1.0" encoding="UTF-8"?>
<module xmlns="urn:jboss:module:1.0" name="com.mysql">
    <resources>
        <resource-root path="mysql-connector-java-5.1.36-bin.jar"/>
    </resources>
    <dependencies>
        <module name="javax.api"/>
        <module name="javax.transaction.api"/>
    </dependencies>
</module>
```

Cài đặt Sqlite JDBC Module

Tạo cấu trúc thư mục như hình. Copy xerial sqlite jdbc driver (<http://xerial.org>) vào.



Nội dung của module.xml như sau

```
<module xmlns="urn:jboss:module:1.1" name="org.xerial">
    <resources>
        <resource-root path="sqlite-jdbc-3.16.1.jar"/>
    </resources>
    <dependencies>
        <module name="javax.api"/>
        <module name="javax.transaction.api"/>
    </dependencies>
</module>
```

Khai báo drivers và datasource

Phân đoạn drivers trong standalone-full.xml có nội dung như sau

```
<drivers>
    <driver name="h2" module="com.h2database.h2">
        <xa-datasource-class>org.h2.jdbc.JdbcDataSource</xa-datasource-class>
    </driver>
    <driver name="sqlserver" module="com.microsoft">
        <xa-datasource-class>
            com.microsoft.sqlserver.jdbc.SQLServerXADataSource </xa-datasource-class>
        </driver>
        <driver name="mysql" module="com.mysql">
            <xa-datasource-class>
                com.mysql.jdbc.optional.MysqlXADataSource</xa-datasource-class>
            </driver>
            <driver name="sqlite" module="org.xerial">
                <xa-datasource-class>org.sqlite.JDBC</xa-datasource-class>
            </driver>
        </drivers>
```

Các phân đoạn datasource như sau

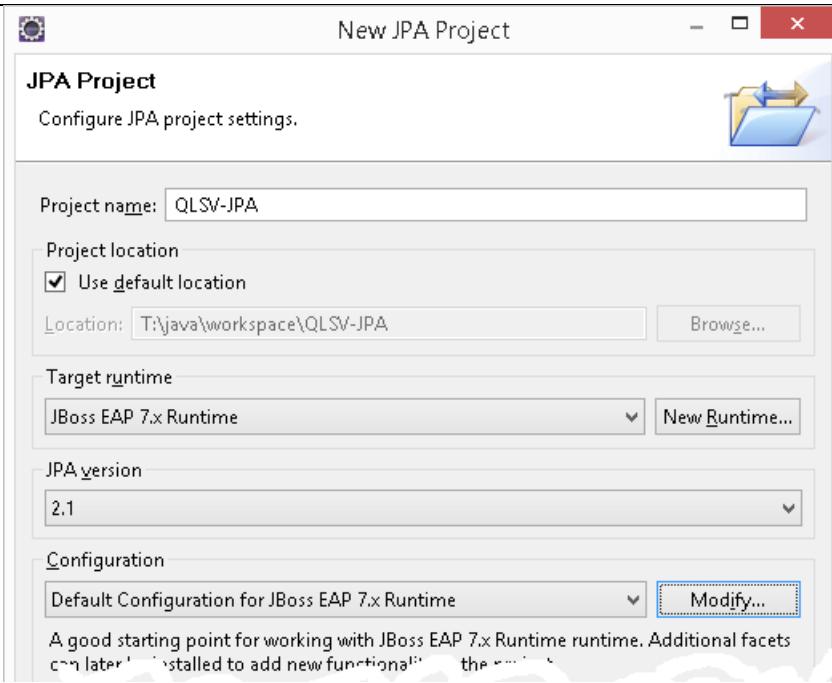
```
<datasource jndi-name="java:/qlsv" pool-name="qlsv" enabled="true" use-java-context="true">
    <connection-url>
        jdbc:sqlserver://localhost:1433;databaseName=qlsv</connection-url>
        <driver>sqlserver</driver>
        <security>
            <user-name>sa</user-name>
            <password>12345678</password>
        </security>
    </datasource>
    <datasource jndi-name="java:jboss/MySqlDS" pool-name="MySqlDS" enabled="true">
        <connection-url>jdbc:mysql://localhost:3306/cdcols</connection-url>
        <driver>mysql</driver>
        <security>
            <user-name>root</user-name>
        </security>
    </datasource>
    <datasource jndi-name="java:/testSqlite" pool-name="testSqlite" enabled="true" use-java-context="true">
        <connection-url>jdbc:sqlite:${jboss.server.data.dir}\xyz.db3</connection-url>
        <driver>sqlite</driver>
    </datasource>
```

JPA PROJECT

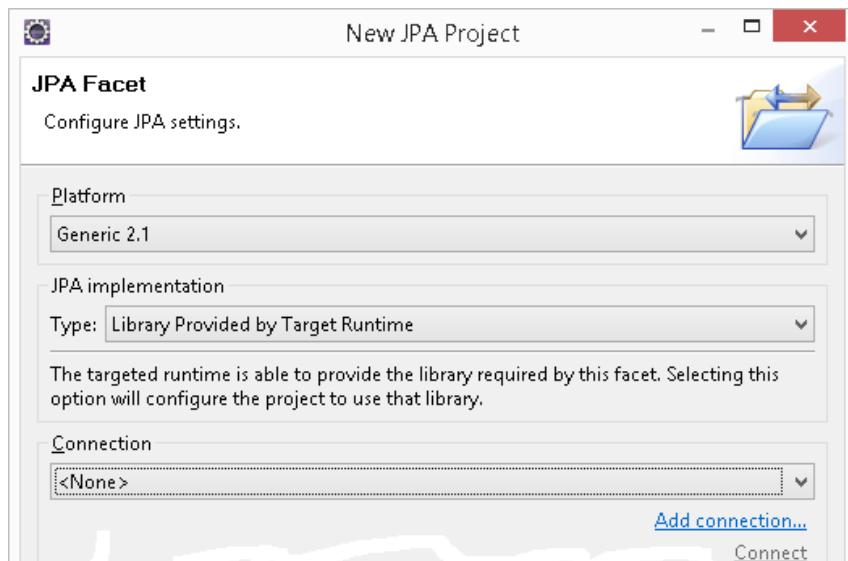
Yêu cầu: Mỗi sinh viên đăng ký nhiều môn học, mỗi môn học được đăng ký bởi nhiều sinh viên. Thông tin của mỗi sinh viên bao gồm: mã số, họ tên, địa chỉ. Thông tin môn học bao gồm: mã số, tên môn học. Kết quả của việc sinh viên đăng ký 1 môn học được phản ánh bằng điểm của sinh khi thi môn học đó.

Tạo một dự án JPA

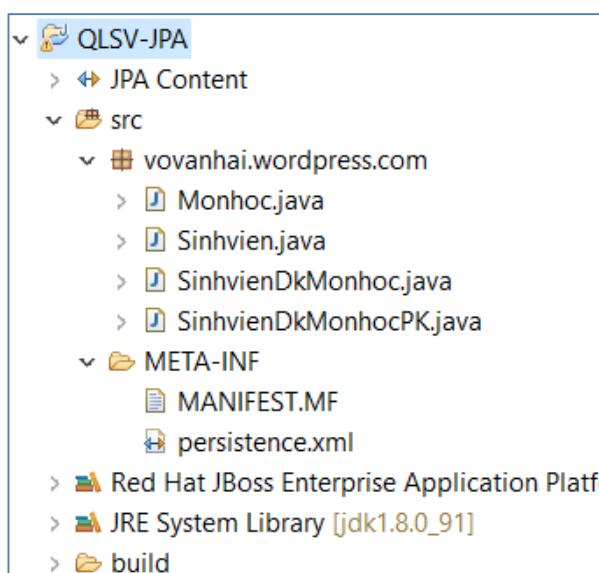




Nhấn next 2 lần



Ta được cấu trúc project như sau:



Tạo lớp mô tả cho sinh viên, kết quả như sau

```

package vovanhai.wordpress.com;
import java.io.Serializable;
import java.util.List;

import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.Id;
import javax.persistence.NamedQueries;
import javax.persistence.NamedQuery;
import javax.persistence.OneToMany;

@Entity
@NamedQueries(value={
    @NamedQuery(name="Sinhvien.findAll", query="select sv from Sinhvien sv")
    ,@NamedQuery(name="Sinhvien.getAllMonhoc", query="select dk from SinhvienDkMonhoc dk where dk.pk=:khoachinhdkmh")
})
public class Sinhvien implements Serializable{
    @Id
    @Column(name="mssv")
    private long mssv;
    @Column(name="hoten", length=70, nullable=false)
    private String hoten;
    @Column(name="diachi", length=200)
    private String diachi;

    @OneToMany(mappedBy = "sv")
    private List<SinhvienDkMonhoc> dssvdkmh;

    public Sinhvien() {
    }
    public Sinhvien(long mssv, String hoten, String diachi) {
        this.mssv = mssv;
        this.hoten = hoten;
        this.diachi = diachi;
    }

    public List<SinhvienDkMonhoc> getDssvdkmh() {
        return dssvdkmh;
    }
    public void setDssvdkmh(List<SinhvienDkMonhoc> dssvdkmh) {
        this.dssvdkmh = dssvdkmh;
    }
    public long getMssv() {
        return mssv;
    }
    public void setMssv(long mssv) {
        this.mssv = mssv;
    }
    public String getHoten() {
        return hoten;
    }
    public void setHoten(String hoten) {
        this.hoten = hoten;
    }
    public String getDiachi() {
        return diachi;
    }
    public void setDiachi(String diachi) {
        this.diachi = diachi;
    }
}

```

```
[}
```

Lớp Môn học

```
package vovanhai.wordpress.com;
import java.io.Serializable;
import java.util.List;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.FetchType;
import javax.persistence.Id;
import javax.persistence.OneToMany;
import javax.persistence.Table;

@Entity
@Table(name="Monhoc")
public class Monhoc implements Serializable{
    @Id
    @Column(name="mamh", length=10, nullable=false)
    private String mamh;

    @Column(name="tenmh", length=200, nullable=false)
    private String tenmh;

    @OneToMany(mappedBy="monhoc", fetch=FetchType.EAGER)
    private List<SinhvienDkMonhoc>dsmhdksv;

    public Monhoc() {
    }
    public Monhoc(String mamh, String tenmh) {
        this.mamh = mamh;
        this.tenmh = tenmh;
    }
    public String getMamh() {
        return mamh;
    }
    public List<SinhvienDkMonhoc> getDsmhdksv() {
        return dsmhdksv;
    }
    public void setDsmhdksv(List<SinhvienDkMonhoc> dsmhdksv) {
        this.dsmhdksv = dsmhdksv;
    }
    public void setMamh(String mamh) {
        this.mamh = mamh;
    }
    public String getTenmh() {
        return tenmh;
    }
    public void setTenmh(String tenmh) {
        this.tenmh = tenmh;
    }
}
```

Giữa hai đối tượng môn học và sinh viên có mối quan hệ nhiều-nhiều nên ta tạo một lớp mô tả cho đối tượng trung gian. Có thể để JPA tự sinh bằng `@ManyToMany` nhưng trong trường hợp này ta tạo đối tượng trung gian có tên `SinhvienDkMonhoc`. Đối tượng này có id là id của 2 đối tượng môn học, sinh viên và một thuộc tính điểm. Code của lớp khóa chính và code của lớp như sau

```
package vovanhai.wordpress.com;
import java.io.Serializable;
import javax.persistence.Embeddable;
```

```

@Embeddable
public class SinhvienDkMonhocPK implements Serializable{
    private long mssv;
    private String mamh;

    public SinhvienDkMonhocPK() {
    }
    public SinhvienDkMonhocPK(long mssv, String mamh) {
        this.mssv = mssv;
        this.mamh = mamh;
    }
    public long getMssv() {
        return mssv;
    }
    public void setMssv(long mssv) {
        this.mssv = mssv;
    }
    public String getMamh() {
        return mamh;
    }
    public void setMamh(String mamh) {
        this.mamh = mamh;
    }
    @Override
    public int hashCode() {
        final int prime = 31;
        int result = 1;
        result = prime * result + ((mamh == null) ? 0 : mamh.hashCode());
        result = prime * result + (int) (mssv ^ (mssv >>> 32));
        return result;
    }
    @Override
    public boolean equals(Object obj) {
        if (this == obj)
            return true;
        if (obj == null)
            return false;
        if (getClass() != obj.getClass())
            return false;
        SinhvienDkMonhocPK other = (SinhvienDkMonhocPK) obj;
        if (mamh == null) {
            if (other.mamh != null)
                return false;
        } else if (!mamh.equals(other.mamh))
            return false;
        if (mssv != other.mssv)
            return false;
        return true;
    }
}
package vovanhai.wordpress.com;

import java.io.Serializable;
import javax.persistence.Entity;
import javax.persistence.Id;
import javax.persistence.ManyToOne;
import javax.persistence.Table;

@Entity
@Table(name="SinhvienDkMonhoc")
public class SinhvienDkMonhoc implements Serializable{
    @Id

```

```

private SinhvienDkMonhocPK pk;
private float diem;

@ManyToOne
private Monhoc monhoc;

@ManyToOne
private Sinhvien sv;

public Monhoc getMonhoc() {
    return monhoc;
}
public void setMonhoc(Monhoc monhoc) {
    this.monhoc = monhoc;
}
public Sinhvien getSv() {
    return sv;
}
public void setSv(Sinhvien sv) {
    this.sv = sv;
}
public SinhvienDkMonhoc() {
}
public SinhvienDkMonhoc(SinhvienDkMonhocPK pk, float diem) {
    super();
    this.pk = pk;
    this.diem = diem;
}
public SinhvienDkMonhocPK getPk() {
    return pk;
}
public void setPk(SinhvienDkMonhocPK pk) {
    this.pk = pk;
}
public float getDiem() {
    return diem;
}
public void setDiem(float diem) {
    this.diem = diem;
}
}

```

File persistent.xml có cấu trúc như sau

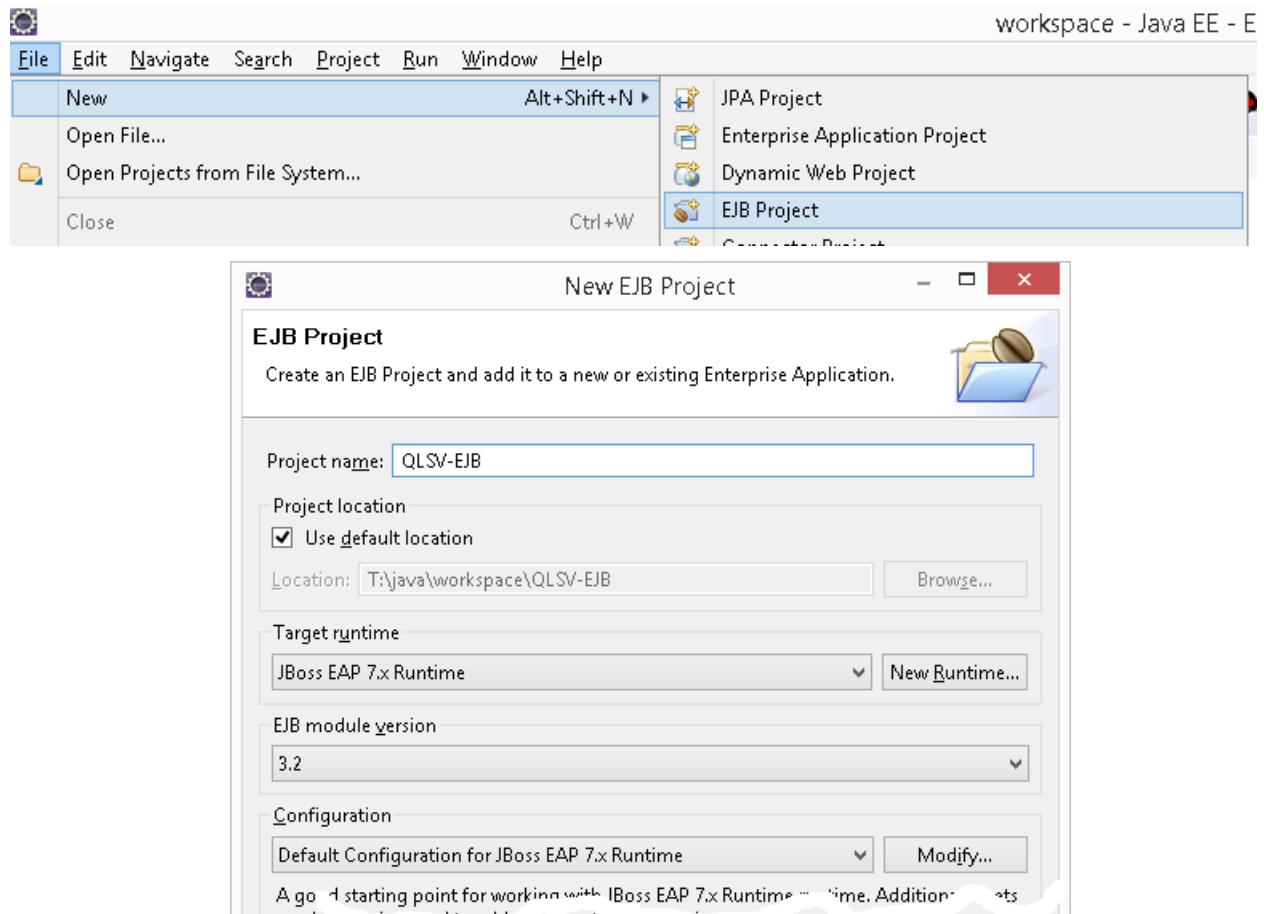
```

<?xml version="1.0" encoding="UTF-8"?>
<persistence      version="2.1"      xmlns="http://xmlns.jcp.org/xml/ns/persistence"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/persistence
   http://xmlns.jcp.org/xml/ns/persistence/persistence_2_1.xsd">
    <persistence-unit name="QLSV-JPA">
        <jta-data-source>java:/qlsv</jta-data-source>
        <class>vovanhai.wordpress.com.Monhoc</class>
        <class>vovanhai.wordpress.com.Sinhvien</class>
        <class>vovanhai.wordpress.com.SinhvienDkMonhoc</class>
        <properties>
            <property name="javax.persistence.schema-
generation.database.action" value="drop-and-create"/>
            <property name="javax.persistence.schema-generation.create-
database-schemas" value="true"/>
        </properties>
    </persistence-unit>
</persistence>

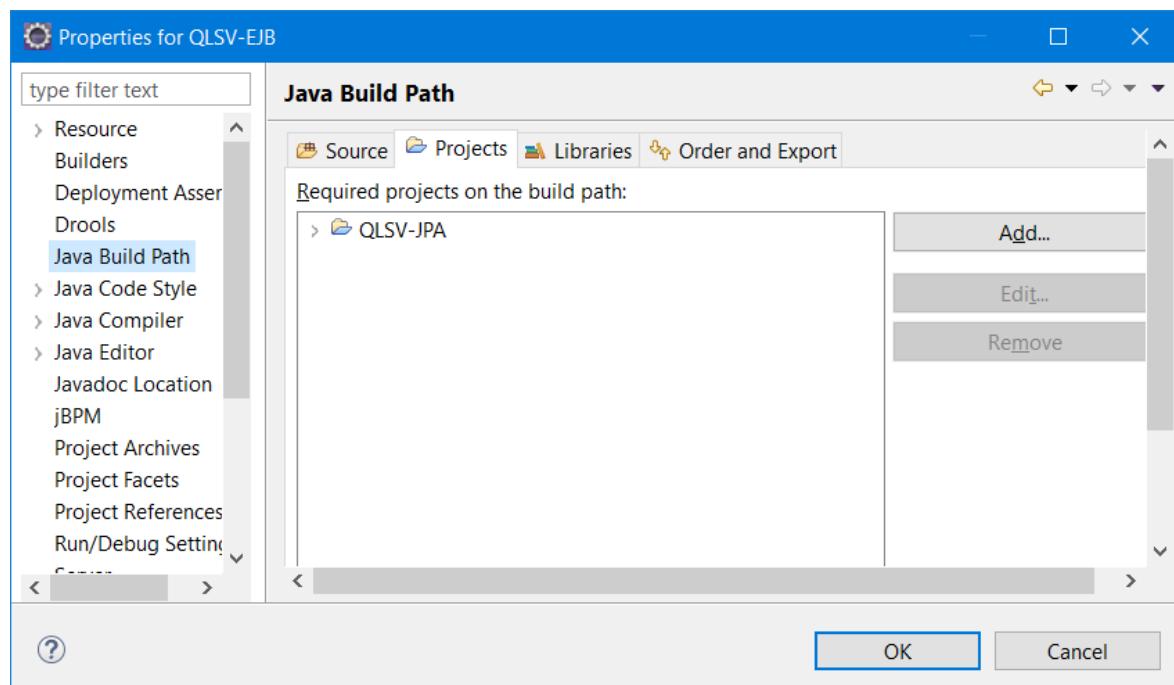
```

EJB PROJECT

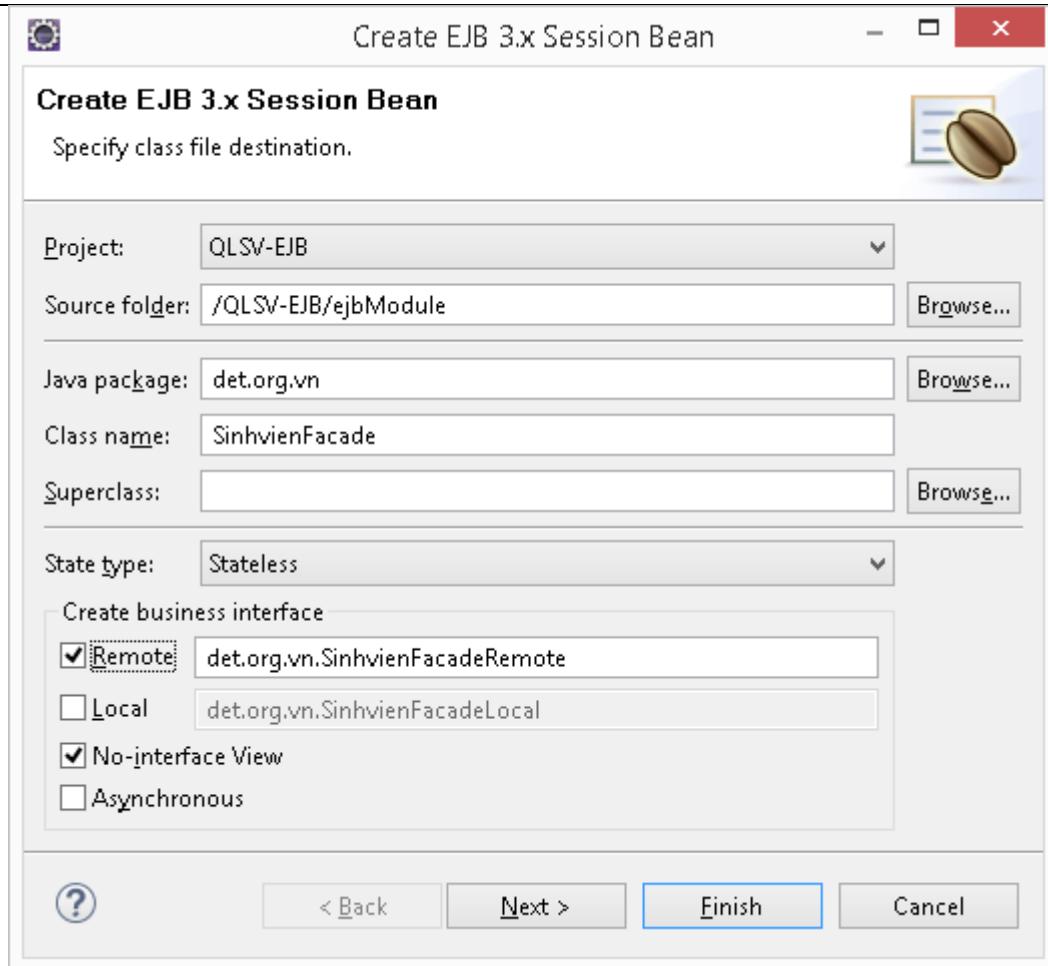
Tạo 1 Stateless SessionBean làm façade cho việc thao tác với JPA



Thêm reference đến JPA project



Tạo façade session bean cho Sinhvien



Test code cho việc thêm 1 sinh viên cũng như lấy về danh sách sinh viên

```
package vovanhai.wordpress.com;

import java.util.List;
import javax.ejb.LocalBean;
import javax.ejb.Stateless;
import javax.persistence.EntityManager;
import javax.persistence.PersistenceContext;
import javax.persistence.TypedQuery;

@Stateless
@LocalBean
public class SinhvienFacade implements SinhvienFacadeRemote {

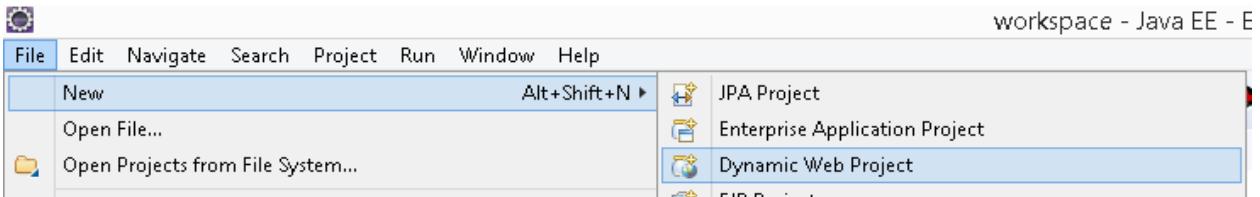
    @PersistenceContext(unitName="QLSV-JPA")
    private EntityManager em;

    public SinhvienFacade() {
    }

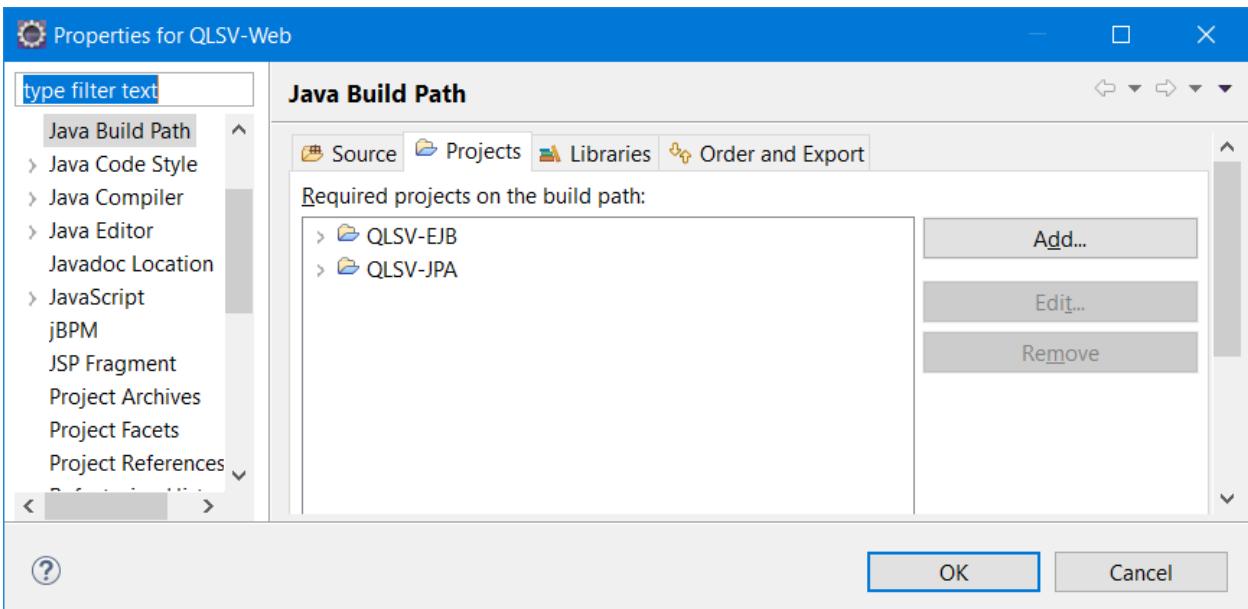
    @Override
    public void insert(Sinhvien sv) {
        em.persist(sv);
    }
    @Override
    public List<Sinhvien> getAllSV() {
        TypedQuery<Sinhvien> q = em.createNamedQuery ("Sinhvien.findAll",
        Sinhvien.class);
        return q.getResultList();
    }
}
```

```
}
```

CLIENT PROJECT



Thêm tham chiếu cho client



Tạo trang index.jsp

```
<%@ page language="java" pageEncoding="UTF-8" %>
<html>
<head>
<title>Index</title>
</head>
<body>
    <form action="InsertSvServlet">
        Mã số:<input name="ms"/><br/>
        Họ tên:<input name="ht"/><br/>
        Địa chỉ:<input name="dc"/><br/>
        <input type="submit" value="Insert"/>
    </form>
</body>
</html>
```

Code cho InsertServlet

```
package vovanhai.wordpress.com;

import java.io.IOException;
import javax.ejb.EJB;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import vovanhai.wordpress.com.Sinhvien;
```

```

import vovanhai.wordpress.com.SinhvienFacadeRemote;

@WebServlet("/InsertSvServlet")
public class InsertSvServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;

    @EJB
    private SinhvienFacadeRemote sr;
    public InsertSvServlet() {
        super();
    }

    protected void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
        long ms=Long.parseLong(request.getParameter("ms"));
        String ht=request.getParameter("ht");
        String dc=request.getParameter("dc");
        Sinhvien sv=new Sinhvien(ms, ht, dc);
        sr.insert(sv);
        response.sendRedirect("ListingSvServlet");
    }
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
        doGet(request, response);
    }
}

```

Code cho ListingSvServlet

```

package vovanhai.wordpress.com;

import java.io.IOException;
import java.io.PrintWriter;
import java.util.List;
import javax.ejb.EJB;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import vovanhai.wordpress.com.Sinhvien;
import vovanhai.wordpress.com.SinhvienFacadeRemote;

@WebServlet("/ListingSvServlet")
public class ListingSvServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;
    @EJB
    private SinhvienFacadeRemote sr;

    public ListingSvServlet() {
        super();
    }

    protected void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
        List<Sinhvien> lst = sr.getAllSV();
        PrintWriter out=response.getWriter();
        out.println("<h1>DANH SÁCH SINH VIÊN</h1>");
        out.println("<table border='1'><tr><th>mã số</th><th> họ tên</th><th>địa
chi</th></tr></table>");
        for(Sinhvien s:lst)
            out.println("<tr><td>" + s.getMssv() + "</td><td>" +
            s.getTen() + "</td><td>" + s.getDiaChi() + "</td></tr>");
    }
}

```

```

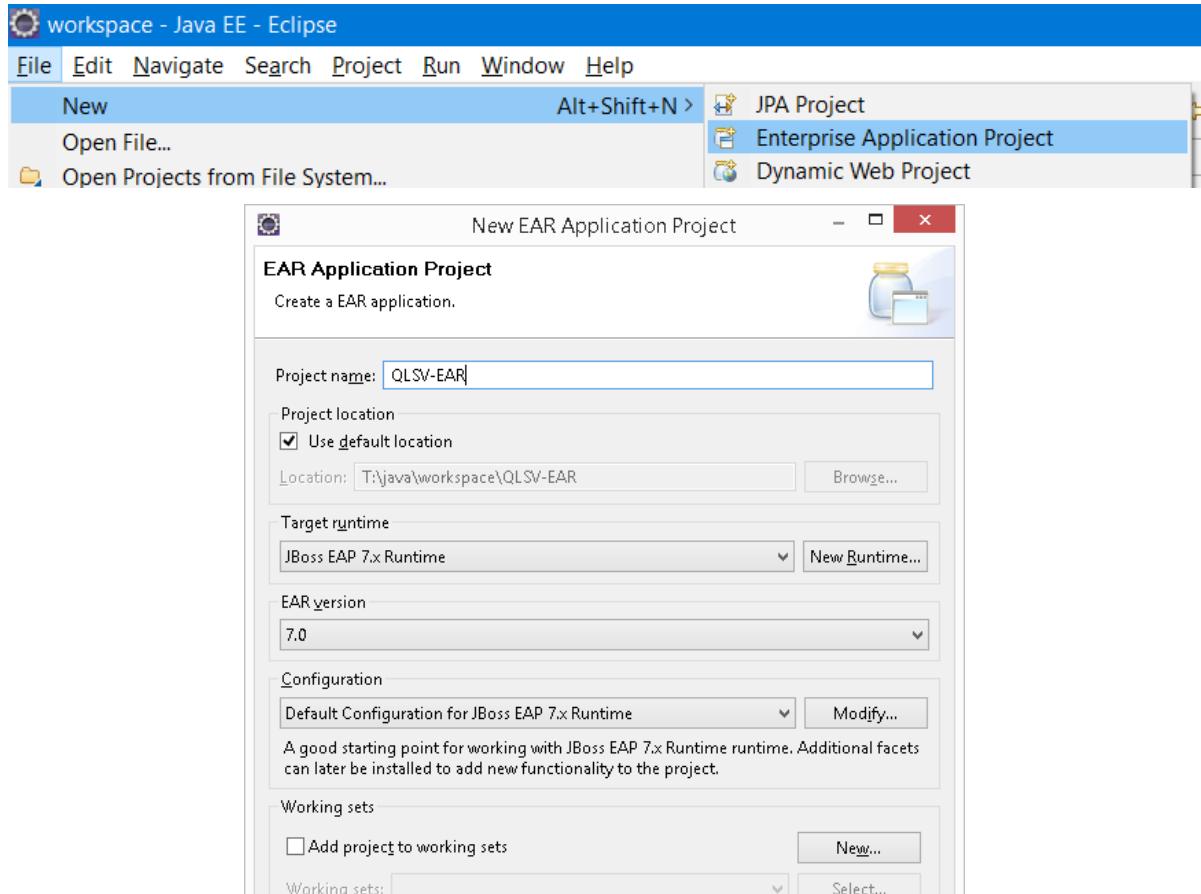
        +s.getHoten()+"</td><td>"+s.getDiachi()+"</td></tr>");
    }

    protected void doPost(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
    doGet(request, response);
}
}

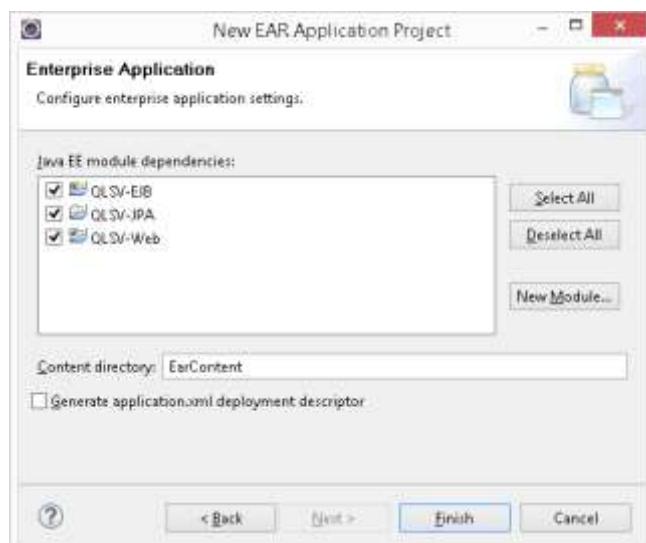
```

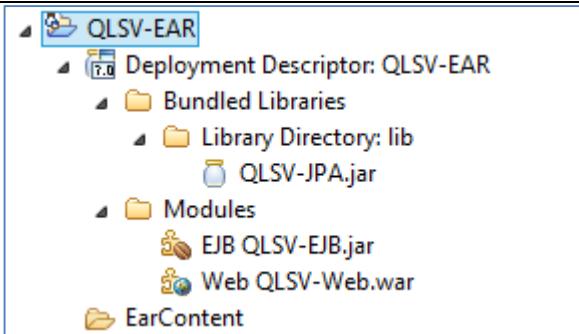
ENTERPRISE APPLICATION PROJECT

Tạo 1 Enterprise Application Project



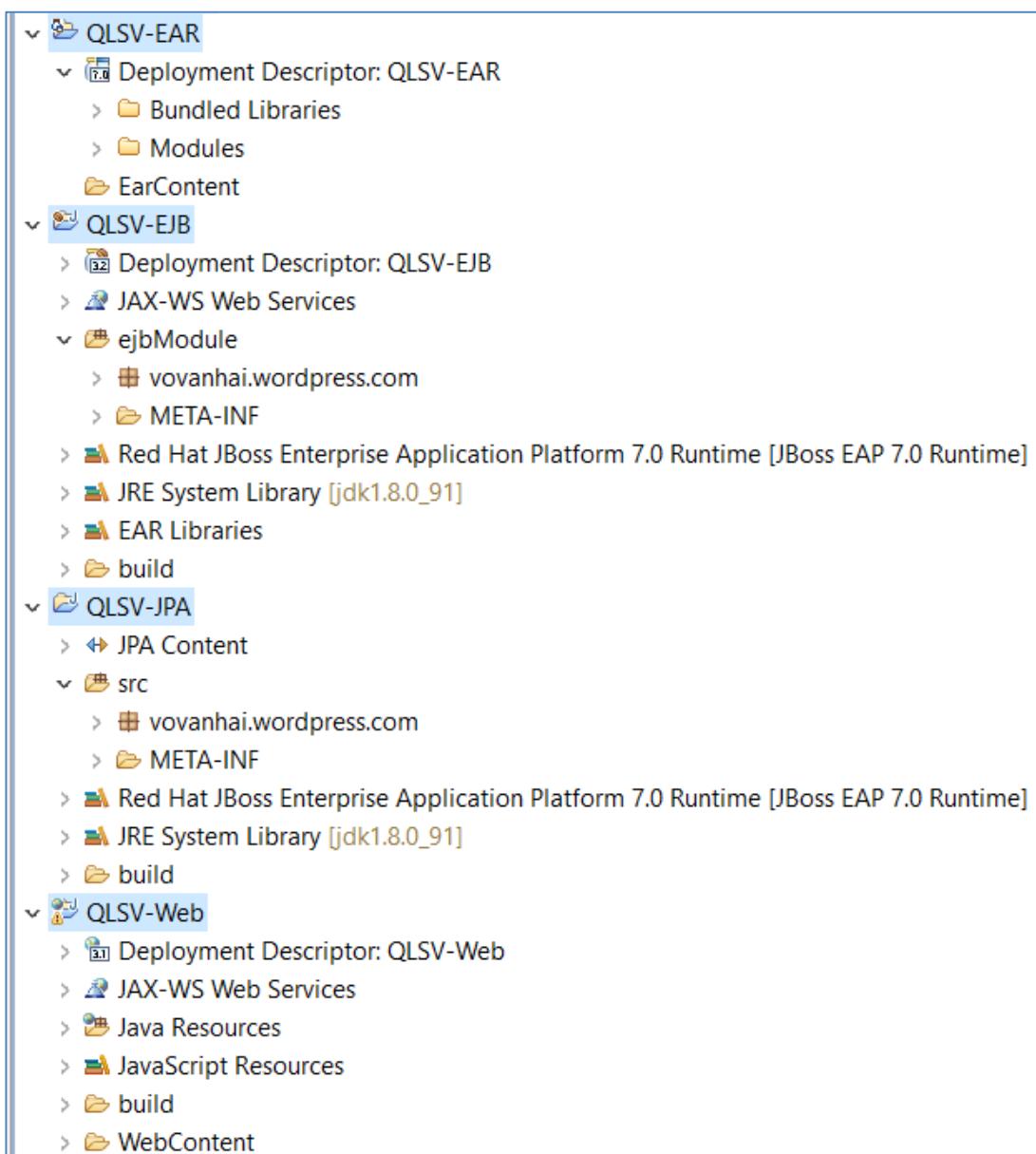
Chọn thêm các module vào eap





TRIỂN KHAI

Cấu trúc của cá ứng dụng như sau



Triển khai EAP lên JBossEAP ta được

```

22:28:38,591 INFO [org.jboss.weld.deployer] (MSC service thread 1-5) WFLYWELD0005: Processing weld deployment QLSV-EJB.jar
22:28:38,605 INFO [org.jboss.as.ejb3.deployment] (MSC service thread 1-5) WFLYEJB00473: JNDI bindings for session bean named 'Sinhvien'
java:global/QLSV-EAR/QLSV-EJB/SinhvienFacade!vovanhai.wordpress.com.SinhvienFacadeRemote
java:app/QLSV-EJB/SinhvienFacade!vovanhai.wordpress.com.SinhvienFacadeRemote
java:jboss/exported/QLSV-EAR/QLSV-EJB/SinhvienFacade!vovanhai.wordpress.com.SinhvienFacadeRemote
java:global/QLSV-EAR/QLSV-EJB/SinhvienFacade!vovanhai.wordpress.com.SinhvienFacade
java:app/QLSV-EJB/SinhvienFacade!vovanhai.wordpress.com.SinhvienFacade
java:module/SinhvienFacade!vovanhai.wordpress.com.SinhvienFacade

22:28:38,606 INFO [org.jboss.as.ejb3.deployment] (MSC service thread 1-5) WFLYEJB00473: JNDI bindings for session bean named 'Monhoc'
java:global/QLSV-EAR/QLSV-EJB/MonhocFacade!vovanhai.wordpress.com.MonhocFacade
java:app/QLSV-EJB/MonhocFacade!vovanhai.wordpress.com.MonhocFacade
java:module/MonhocFacade!vovanhai.wordpress.com.MonhocFacade
java:global/QLSV-EAR/QLSV-EJB/MonhocFacade
java:app/QLSV-EJB/MonhocFacade
java:module/MonhocFacade

22:28:38,742 INFO [org.jboss.weld.deployer] (MSC service thread 1-4) WFLYWELD0006: Starting Services for CDI deployment: QLSV-EAR.jar
22:28:38,759 INFO [org.jboss.weld.Version] (MSC service thread 1-4) WELD-000900: 2.3.3 (redhat)

```

Index

http://localhost:8080/QLSV-Web/

Mã số:	1001
Họ tên:	Nguyễn văn Tèo
Địa chỉ:	Nguyễn văn bão, f4, GV
Insert	

http://localhost:8080/QLSV-Web/ListingSvServlet

DANH SÁCH SINH VIÊN

mã s?	h? tên	??a ch?
1001	Nguy?n Van Tèo	12 nguy?n van b?o, GV
1003	Truong Van Tun	43 Nguy?n Ki?m GV

BÀI TẬP

Quản lý dự án

Cho một mô tả logic như sau:

Một nhân viên (Employee) có một địa chỉ (Address) duy nhất.

Một Address chỉ thuộc về một nhân viên.

Một Employee thuộc về một phòng ban (Department)

Một Department có nhiều nhân viên.

Một dự án (Project) có nhiều nhân viên tham gia

Một nhân viên có thể tham gia nhiều dự án

Trong đó các thông tin cơ sở của các entity như sau:

Employee (mã số, họ tên, ngày sinh, số điện thoại)

Address (mã số, số nhà, đường, phường, quận)

Department (id, tên phòng ban, mô tả)

Project (id, tên project, ngày bắt đầu, ngày kết thúc)

Hãy dùng JPA để ánh xạ các entities trên. Viết các lớp controller cho các entities này.

Viết chương trình cho phép thao tác với thành phần vừa tạo.

Chương trình trắc nghiệm

Cho một mô tả sau:

Một thư viện câu hỏi trắc nghiệm gồm một tập các câu hỏi trắc nghiệm,

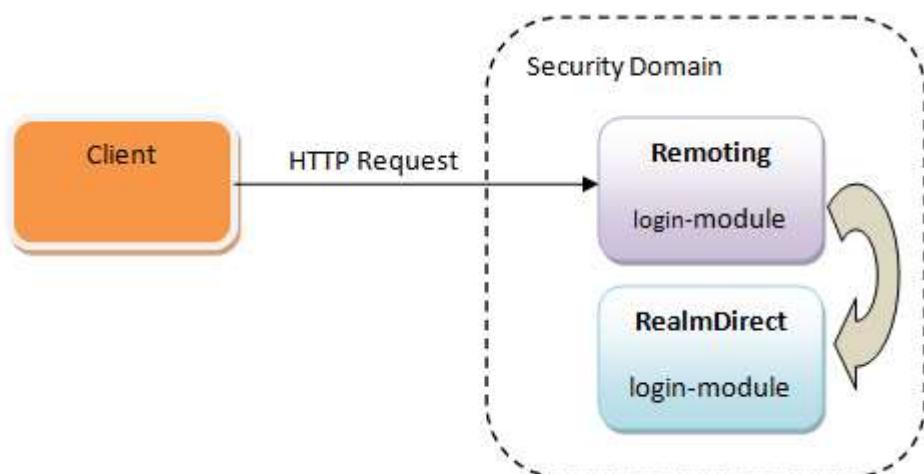
Mỗi câu hỏi gồm một định danh để xác định, nội dung của câu hỏi, và độ khó của nó (bốn mức độ: dễ, trung bình, khó, rất khó)

Mỗi câu hỏi có một danh sách các đáp án, mỗi đáp án có một định danh, nội dung của đáp án và một xác định xem đó có phải là đáp án đúng không.

(mỗi câu hỏi có thể có nhiều lựa chọn và có thể chọn nhiều đáp áp đúng)

- a) Hãy xây dựng các entities để mô tả các thông tin trên sử dụng JPA
- b) Hãy xây dựng chương trình trắc nghiệm đơn giản.

VII. EJB SECURITY



CẤU HÌNH

AUTHENTICATION MODULE

Simple Login Molue

Có thể dùng chung security-domain có tên **other** của JBossEAP. Dùng user của application-users.properties

NÊN DÙNG CÁCH NÀY

```
<security-domain name="other" cache-type="default">
    <authentication>
        <login-module code="Remoting" flag="optional">
            <module-option name="password-stacking" value="useFirstPass"/>
        </login-module>
        <login-module code="RealmDirect" flag="required">
            <module-option name="password-stacking" value="useFirstPass"/>
        </login-module>
    </authentication>
</security-domain>
```

Application roles trong ví dụ sau được thêm vào như sau

hai=users,role2 ty=role1

Dùng chung với Server Module

Cập nhật security-realms như sau trong **standalone-full.xml**

```
<security-realms>
    <security-realm name="ManagementRealm">
        <authentication>
            <local default-user="$local" skip-group-loading="true"/>
            <properties path="mgmt-users.properties" relative-to="jboss.server.config.dir"/>
        </authentication>
        <authorization map-groups-to-roles="true">
            <properties path="mgmt-groups.properties" relative-to="jboss.server.config.dir"/>
        </authorization>
    </security-realm>
    <security-realm name="ApplicationRealm">
        <authentication>
            <local default-user="$local" allowed-users="*" skip-group-loading="true"/>
            <properties path="application-users.properties" relative-to="jboss.server.config.dir"/>
        </authentication>
        <authorization map-groups-to-roles="true">
            <properties path="application-roles.properties" relative-to="jboss.server.config.dir"/>
        </authorization>
    </security-realm>
</security-realms>
```

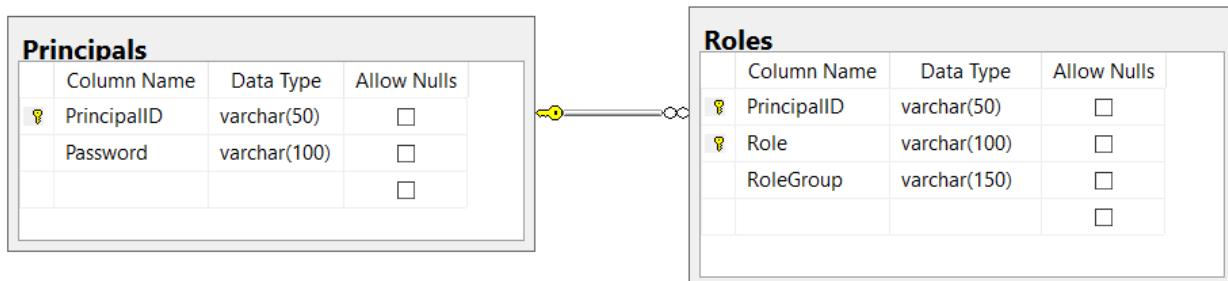
Thêm phân đoạn security-domain có tên management vào standalone-full.xml

```
<security-domain name="management" cache-type="default">
    <authentication>
        <login-module code="RealmDirect" flag="required">
            <module-option name="realm" value="ManagementRealm"/>
        </login-module>
    </authentication>
</security-domain>
```

JAAS JDBC Login Module

Tạo cơ sở dữ liệu có tên **SecureLoginModuleJBossEAP**

Cấu trúc như sau



Thêm một user và 2 roles vào

```
INSERT INTO Principals VALUES('vovanhai', 'hoilamgi')
INSERT INTO Roles VALUES('vovanhai', 'manager', 'manager')
INSERT INTO Roles VALUES('vovanhai', 'caller', 'CallerPrincipal')
```

Thêm datasource

```
<datasource jndi-name="java:/dbLoginModuleDs" pool-name="dbLoginModuleDs"
    enabled="true" use-java-context="true">
    <connection-url>
        jdbc:sqlserver://localhost:1433;databaseName=SecureLoginModuleJBossEAP
    </connection-url>
    <driver>sqlserver</driver>
    <security>
        <user-name>sa</user-name>
        <password>12345678</password>
    </security>
</datasource>
```

Config security domain standalone-full.xml

```
<security-domain name="vvh-security-domain" cache-type="default">
    <authentication>
```

```

<login-module
"org.jboss.security.auth.spi.DatabaseServerLoginModule" flag="optional">
<module-option name="dsJndiName" value="java:/dbLoginModuleDs"/>
<module-option name="principalsQuery" value="select Password from
Principals where PrincipalID=?"/>
<module-option name="rolesQuery" value="Select Role, RoleGroup from Roles
where PrincipalID = ?"/>
</login-module>
</authentication>
</security-domain>

```

EJB AUTHORIZATION

Chúng ta lưu ý đến một số annotation trong ejb

@DeclareRoles	Indicates that class will accept those declared roles. Annotations are applied at class level.
@RolesAllowed	Indicates that a method can be accessed by user of role specified. Can be applied at class level resulting which all methods of class can be accessed by user of role specified
@DenyAll	Indicates that business method is not accessible to any of user specified at class or at method level
@PermitAll	Indicates that business method is accessible to all. Can be applied at class as well as at method level
@SecurityDomain	Chỉ định security-domain nào sẽ được dùng để authenticate-authorize cho bean
@DeclareRoles	Khai báo các roles để cấp quyền cho các method có trong bean

EJB PROJECT

Tạo một EJB project có tên **security-ejb**

Tạo một session-bean có tên Calculator gồm một số phương thức demo cho việc cấp quyền



Code của Calculator.java như sau

```

package vovanhai.wordpress.com;

import java.security.Principal;
import javax.annotation.Resource;
import javax.annotation.security.DeclareRoles;

```

```

import javax.annotation.security.DenyAll;
import javax.annotation.security.PermitAll;
import javax.annotation.security.RolesAllowed;
import javax.ejb.LocalBean;
import javax.ejb.SessionContext;
import javax.ejb.Stateless;
import org.jboss.security.annotation.SecurityDomain;

@Stateless
@LocalBean
@SecurityDomain("other") //dùng security-domain có tên other
@DeclareRoles({"role1","role2"}) //các role của users
public class Calculator {
    @Resource
    private SessionContext ctx;

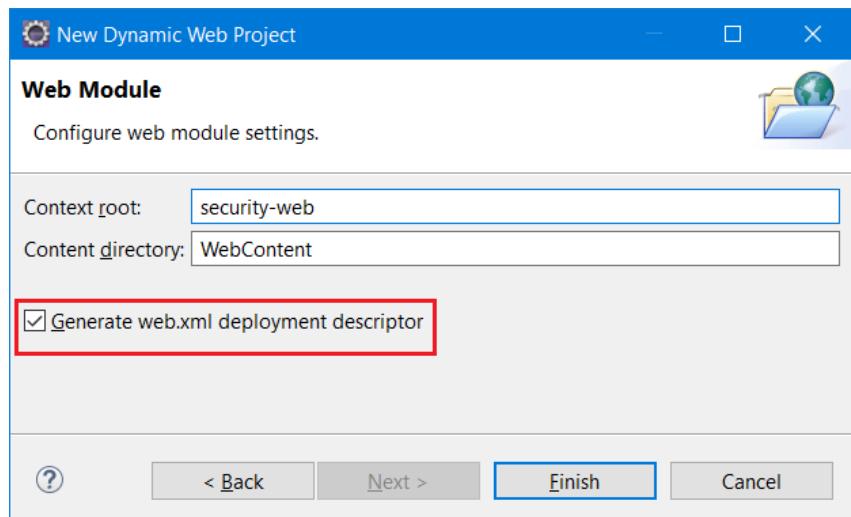
    public Calculator() {
    }
    @RolesAllowed("role1") //cấp quyền cho role1
    public int cong(int a,int b){
        return a+b;
    }
    @RolesAllowed("role2")//cấp quyền cho role2
    public int tru(int a,int b){
        return a-b;
    }
    @DenyAll ////cấm tất cả
    public int nhan(int a,int b){
        return a*b;
    }

    @PermitAll //cho phép tất cả
    public String getSecurityInfo() {
        // Session context injected using the resource annotation
        Principal principal = ctx.getCallerPrincipal();
        return principal.toString();
    }
}

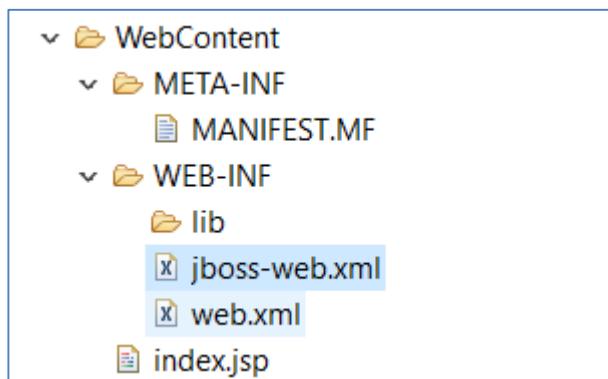
```

CLIENT

Trong Dynamic Web Project có tên security-web, khi tạo project ta nên nhấn chọn vào để sinh ra file web.xml (nhấn next 2 lần)



Sau đó trong thư mục WEB-INF của project, ta thêm một file có tên jboss-web.xml. Kết quả như hình



Nội dung file jboss-web.xml như sau

```
jboss-web.xml
1 <?xml version="1.0" encoding="UTF-8"?>
2 <jboss>
3 <!-- All bean containers use this security manager by default -->
4 <security-domain>management</security-domain>
5 </jboss>
```

Nội dung file web.xml như sau

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://xmlns.jcp.org/xml/ns/javaee"
  xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
  http://xmlns.jcp.org/xml/ns/javaee/web-app_3_1.xsd"
  id="WebApp_ID" version="3.1">
  <display-name>security-web</display-name>
  <welcome-file-list>
    <welcome-file>index.html</welcome-file>
    <welcome-file>index.htm</welcome-file>
    <welcome-file>index.jsp</welcome-file>
    <welcome-file>default.html</welcome-file>
    <welcome-file>default.htm</welcome-file>
    <welcome-file>default.jsp</welcome-file>
  </welcome-file-list>
  <security-constraint>
    <web-resource-collection>
      <web-resource-name>Web-resources</web-resource-name>
      <url-pattern>/CalcServlet</url-pattern>
    </web-resource-collection>
    <auth-constraint>
      <role-name>role1</role-name>
      <role-name>role2</role-name>
    </auth-constraint>
  </security-constraint>
  <security-role>
    <role-name>role1</role-name>
  </security-role>
  <login-config>
    <auth-method>BASIC</auth-method>
    <realm-name>other</realm-name>
  </login-config>
</web-app>
```

Ở đây có 4 loại login-config

- Basic authentication
- Form-based authentication
- Digest authentication
- Client authentication
- Mutual authentication

(Xem thêm ở <http://vovanhai.wordpress.com/>)

Ngoài ra ta có thể chỉ định thêm Secure Connection (CONFIDENTIAL, INTEGRAL, hoặc NONE). Tuy nhiên cần cấu hình https.

```
<user-data-constraint>
    <transport-guarantee>CONFIDENTIAL</transport-guarantee>
</user-data-constraint>
```

Thêm trang index.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Calculator Security</title>
</head>
<body>
    <form action="CalcServlet">
        Số A:<input name="a"/><br/>
        Số B:<input name="b"/><br/>
        <input type="submit" name="cong" value="Cộng"/>
        <input type="submit" name="tru" value="Trừ"/>
        <input type="submit" name="nhan" value="Cộng"/>
        <input type="submit" name="infos" value="Get Infos"/>
    </form>
</body>
</html>
```

Servlet cho việc xử lý (chú ý là 1 servlet xử lý cho 4 nút submit)

```
package vovanhai.wordpress.com;

import java.io.IOException;
import java.io.PrintWriter;

import javax.ejb.EJB;
import javax.ejb.EJBAccessException;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

@WebServlet("/CalcServlet")
public class CalcServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;

    @EJB
    private Calculator calc;
    public CalcServlet() {
```

```

        super();
    }

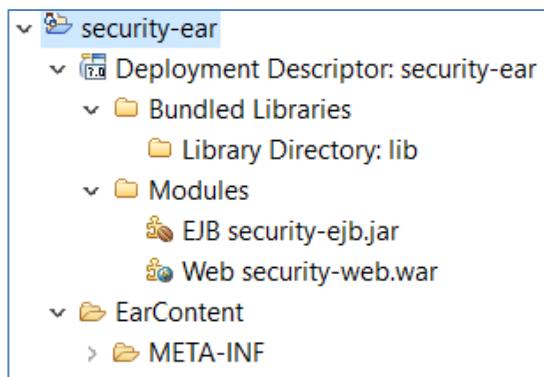
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
    PrintWriter out=response.getWriter();
    try {
        int a=Integer.parseInt(request.getParameter("a"));
        int b=Integer.parseInt(request.getParameter("b"));
        Object cong=request.getParameter("cong");
        Object tru=request.getParameter("tru");
        Object nhan=request.getParameter("nhan");
        Object infos=request.getParameter("infos");
        int kq=0;
        if(infos!=null) {
            String si=calc.getSecurityInfo();
            out.println("<h1>principle: "+si+"</h1>");
        }else{
            if(cong!=null) kq=calc.cong(a, b);
            if(tru!=null) kq=calc.tru(a, b);
            if(nhan!=null) kq=calc.nhan(a, b);
            out.println("<h1>"+kq+"</h1>");
        }
    } catch (NumberFormatException e) {
        out.println("Re-enter: "+e);
    }catch(EJBAccessException se){
        out.println("Security: "+se.getMessage());
    }catch(Exception se){
        out.println("Error: "+se.getMessage());
    }
}

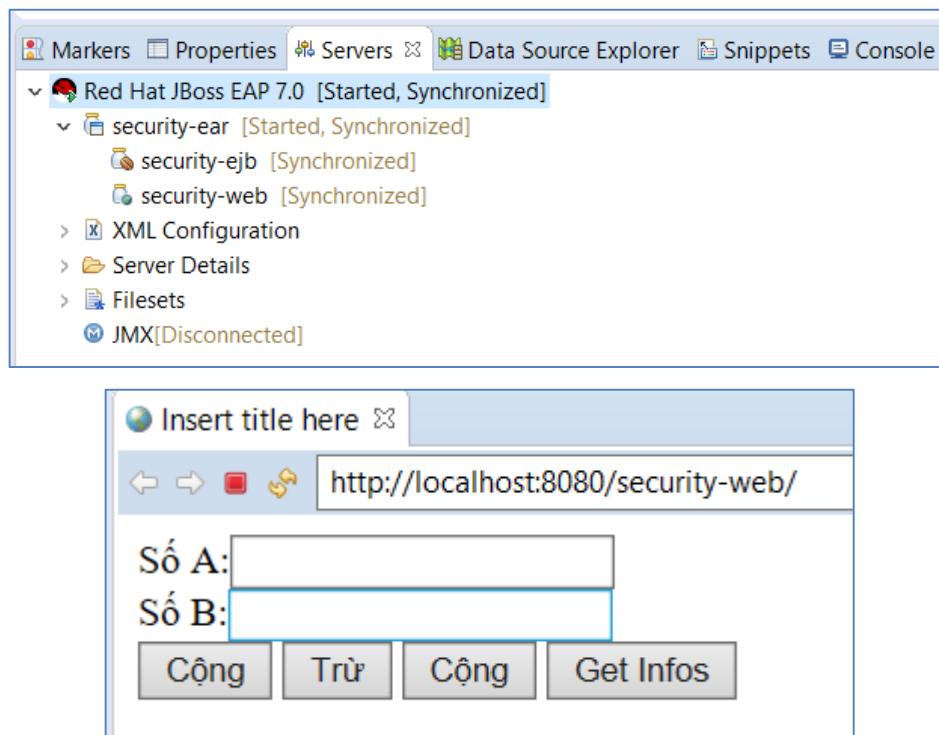
protected void doPost(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
    doGet(request, response);
}
}

```

ENTERPRISE APPLICATION PROJECT

Tạo EAP





HTTPS – SSL**Bước 1: Gen keystore**

```
keytool -genkey -alias myssl -keyalg RSA -keystore t:\java_SA\myssl.keystore
```

password là changeit

```
C:\Windows\system32\cmd.exe
C:\Program Files\Java\jdk1.8.0_101\bin>keytool -genkey -alias myssl -keyalg RSA -keystore t:\java_SA\myssl.keystore
Enter keystore password:
Re-enter new password: password:changeit
They don't match. Try again
Enter keystore password:
Re-enter new password:
What is your first and last name?
[Unknown]: vo van hai
What is the name of your organizational unit?
[Unknown]: fit
What is the name of your organization?
[Unknown]: iuh
What is the name of your City or Locality?
[Unknown]: saigon
What is the name of your State or Province?
[Unknown]: tphcm
What is the two-letter country code for this unit?
[Unknown]: vn
Is CN=vo van hai, OU=fit, O=iuh, L=saigon, ST=tphcm, C=vn correct?
[no]: yes
Enter key password for <myssl>
<RETURN if same as keystore password>
Re-enter new password: password:changeit
```

Ta sẽ được file **myssl.keystore** trong thư mục **t:\java_SA**. Copy file này vào thư mục **%JBOSS_EAP_HOME%\standalone\configuration** như hình:

Name	Date modified	Type	Size
application-roles.properties	4/18/2016 4:54 PM	PROPERTIES File	1 KB
application-users.properties	4/18/2016 4:54 PM	PROPERTIES File	1 KB
logging.properties	4/18/2016 4:54 PM	PROPERTIES File	3 KB
mgmt-groups.properties	4/18/2016 4:54 PM	PROPERTIES File	1 KB
mgmt-users.properties	4/18/2016 4:54 PM	PROPERTIES File	2 KB
myssl.keystore	3/22/2017 1:01 PM	KEYSTORE File	3 KB
standalone.xml	4/18/2016 4:54 PM	XML File	21 KB
standalone-full.xml	4/18/2016 4:54 PM	XML File	24 KB
standalone-full-ha.xml	4/18/2016 4:54 PM	XML File	27 KB
standalone-ha.xml	4/18/2016 4:54 PM	XML File	24 KB

Bước 2: Mở file standalone-full.xml lên, thêm vào các cấu hình sau

- Thêm 1 security-realm có tên SslRealm như sau

```
<security-realm name="SslRealm">
  <server-identities>
    <ssl>
      <keystore path="myssl.keystore" alias="myssl" keystore-password="changeit"
                relative-to="jboss.server.config.dir"/>
    </ssl>
  </server-identities>
</security-realm>
```

```

<security-realms>
    <security-realm name="ManagementRealm">
    <security-realm name="ApplicationRealm">
        <security-realm name="SslRealm">
            <server-identities>
                <ssl>
                    <keystore path="myssl.keystore" alias="myssl" keystore-password="changeit"
                        relative-to="jboss.server.config.dir"/>
                </ssl>
            </server-identities>
        </security-realm>
    </security-realms>

```

- Thêm dòng config ssl sau vào subsystem **undertow**

```

<https-listener name="default-ssl" socket-binding="https" security-
realm="SslRealm"/>

<subsystem xmlns="urn:jboss:domain:undertow:3.1">
    <buffer-cache name="default"/>
    <server name="default-server">
        <http-listener name="default" socket-binding="http" redirect-socket="https"/>
        <https-listener name="default-ssl" socket-binding="https" security-realm="SslRealm"/>
        <host name="default-host" alias="localhost">
            <location name="/" handler="welcome-content"/>
            <filter-ref name="server-header"/>
            <filter-ref name="x-powered-by-header"/>
        </host>
    </server>
    <servlet-container name="default">

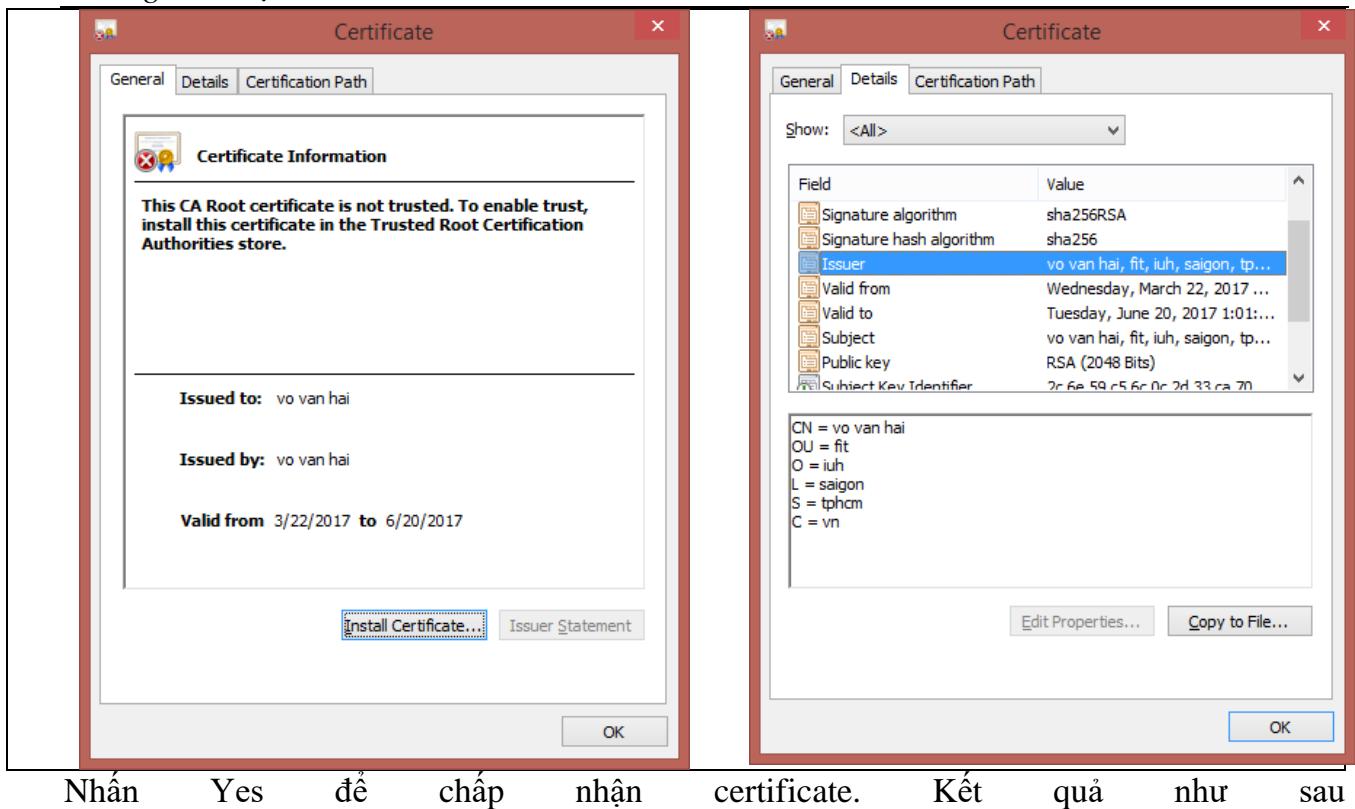
```

Bước 3: Tiến hành start server

Bước 4: Sau đó đăng nhập vào <https://localhost:8443/>



Nhấn View Certificate để xem



Nhân Yes để chấp nhận certificate. Kết quả như sau

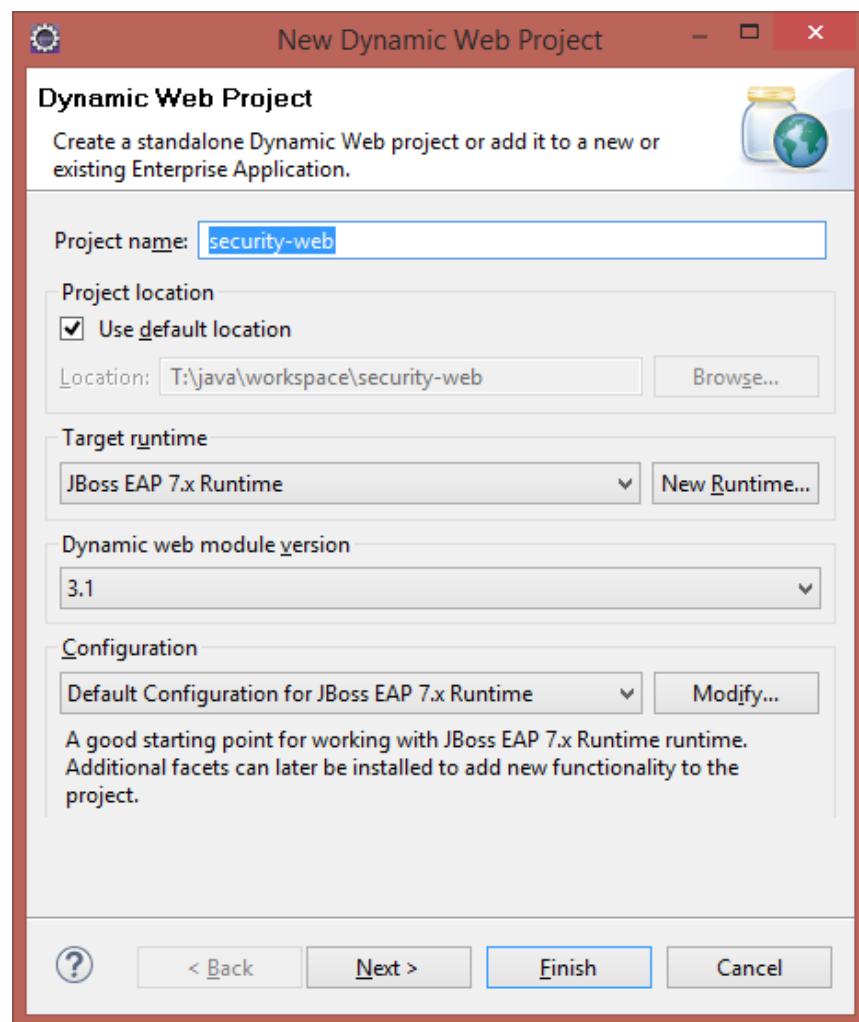
Welcome to JBoss EAP 7

Your Red Hat JBoss Enterprise Application Platform is running.

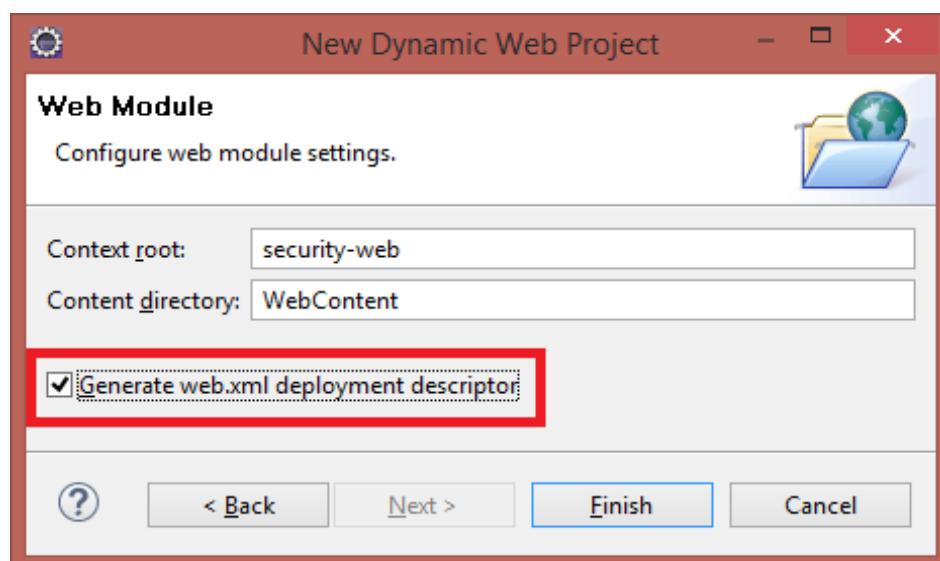
[Administration Console](#) | [Documentation](#) | [Online User Groups](#)

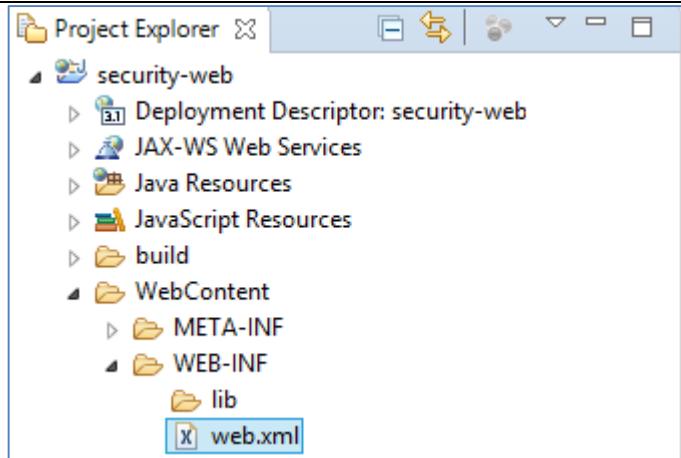
To replace this page simply deploy your own war with / as its context path.

To disable it, remove the "welcome-content" handler for location / in the undertow subsystem.

Bước 5: Tạo Dynamic Web Project

Nhấn Next 2 lần để đảm bảo ta check vào như hình để có web.xml

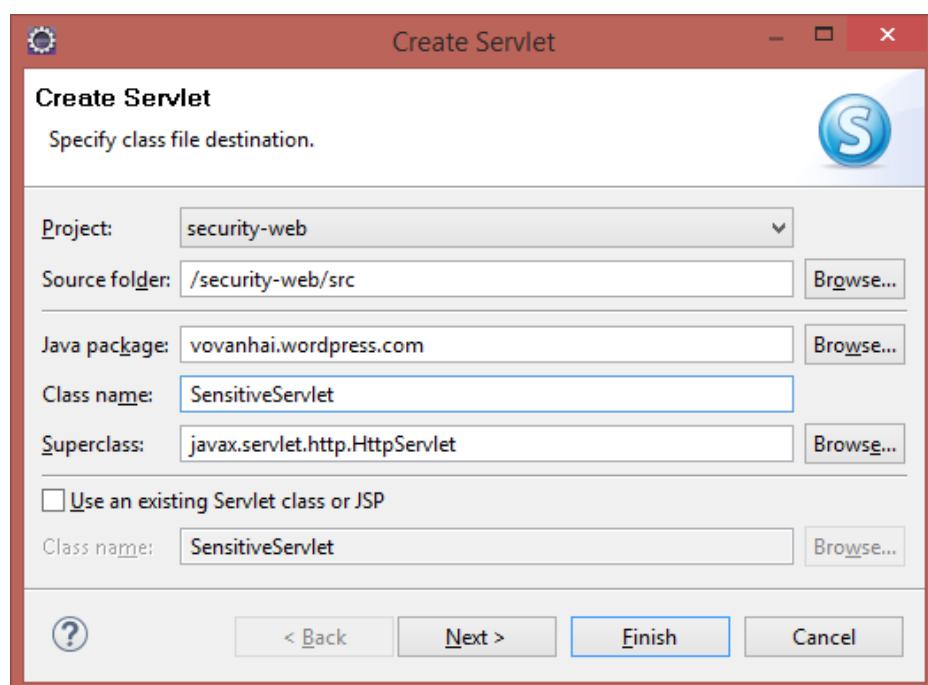


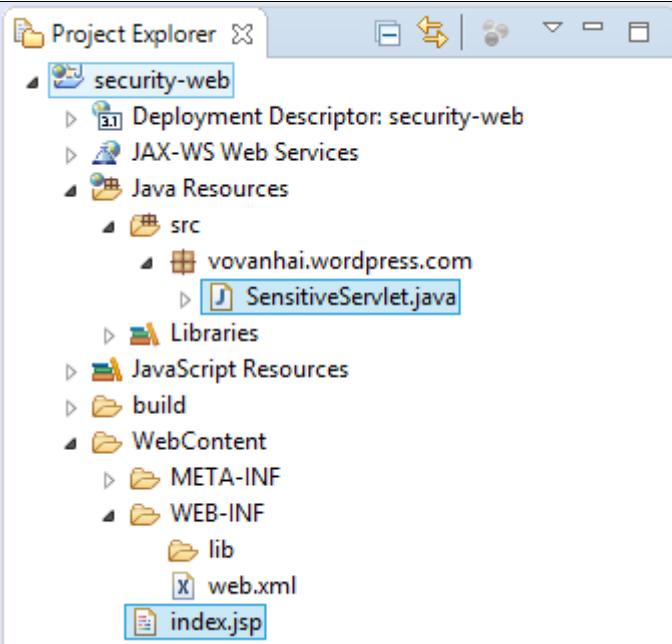


Tạo trang index với nội dung sau

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8"%>
<html>
<head>
<title>Index</title>
</head>
<body>
    <h1>This is sample security</h1>
    <a href="SensitiveServlet">Click here to go to sensitive area</a>
</body>
</html>
```

Tạo 1 servlet có tên SensitiveServlet. Giả sử đây là servlet ta cần bảo vệ





```

package vovanhai.wordpress.com;

import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

@WebServlet("/SensitiveServlet")
public class SensitiveServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;

    public SensitiveServlet() {
        super();
    }

    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        PrintWriter out = response.getWriter();
        out.println("<h1>Sensitive content</h1>");
        out.close();
    }

    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        doGet(request, response);
    }
}

```

Cấu hình web.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns="http://xmlns.jcp.org/xml/ns/javaee"
    xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
    http://xmlns.jcp.org/xml/ns/javaee/web-app_3_1.xsd"
    id="WebApp_ID"
    version="3.1">
    <display-name>security-web</display-name>
    <welcome-file-list>
        <welcome-file>index.html</welcome-file>
        <welcome-file>index.jsp</welcome-file>
    </welcome-file-list>

```

```

<security-constraint>
    <display-name>secure web</display-name>
    <web-resource-collection>
        <web-resource-name>secure test</web-resource-name>
        <url-pattern>/SensitiveServlet</url-pattern>
    </web-resource-collection>
    <auth-constraint>
        <role-name>employee</role-name>
    </auth-constraint>
    <user-data-constraint>
        <description>transfer data guaranty</description>
        <transport-guarantee>CONFIDENTIAL</transport-guarantee>
    </user-data-constraint>
</security-constraint>
<login-config>
    <auth-method>BASIC</auth-method>
    <realm-name>other</realm-name>
</login-config>
</web-app>

```

Giải thích

```

1  <?xml version="1.0" encoding="UTF-8"?>
2  <web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3      xmlns="http://xmlns.jcp.org/xml/ns/javaee"
4      xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee http://xmlns.jcp.org/xml/ns/javaee/web-app_3_1.xsd"
5      id="WebApp_ID" version="3.1">
6      <display-name>security-web</display-name>
7      <welcome-file-list>
8          <welcome-file>index.html</welcome-file>
9          <welcome-file>index.jsp</welcome-file>
10     </welcome-file-list>
11     <security-constraint>
12         <display-name>secure web</display-name>
13         <web-resource-collection>
14             <web-resource-name>secure test</web-resource-name>
15             <url-pattern>/SensitiveServlet</url-pattern> Qui định tài nguyên cần bảo vệ. /* có nghĩa là bảo vệ tất cả
16         </web-resource-collection>
17         <auth-constraint>
18             <role-name>employee</role-name> -- dùng để quy định role nào được cấp quyền truy cập
19         </auth-constraint>
20         <user-data-constraint>
21             <description>transfer data guaranty</description>
22             <transport-guarantee>CONFIDENTIAL</transport-guarantee> -- Quy định cách thức bảo vệ dữ liệu trên
23             </user-data-constraint>
24         </security-constraint>
25     <login-config>
26         <auth-method>BASIC</auth-method> -- phương thức đăng nhập
27         <realm-name>other</realm-name>
28     </login-config>
29 </web-app>

```

<http://vovanhai.wordpress.com>

Cần cấu hình thêm vài bước để có thể chạy

1. Cấu hình trong standalone-full.xml

```

<security-realm name="ApplicationRealm">
    <authentication>
        <local default-user="$local" allowed-users="*" skip-group-loading="true"/>
        <properties path="application-users.properties" relative-to="jboss.server.config.dir"/>
    </authentication>
    <authorization map-groups-to-roles="true">
        <properties path="application-roles.properties" relative-to="jboss.server.config.dir"/>
    </authorization>
</security-realm>

```

2. Thêm người dùng

Chạy file %JBOSS_EAP_HOME%\add-user.bat. Chọn b để thêm Application User

```

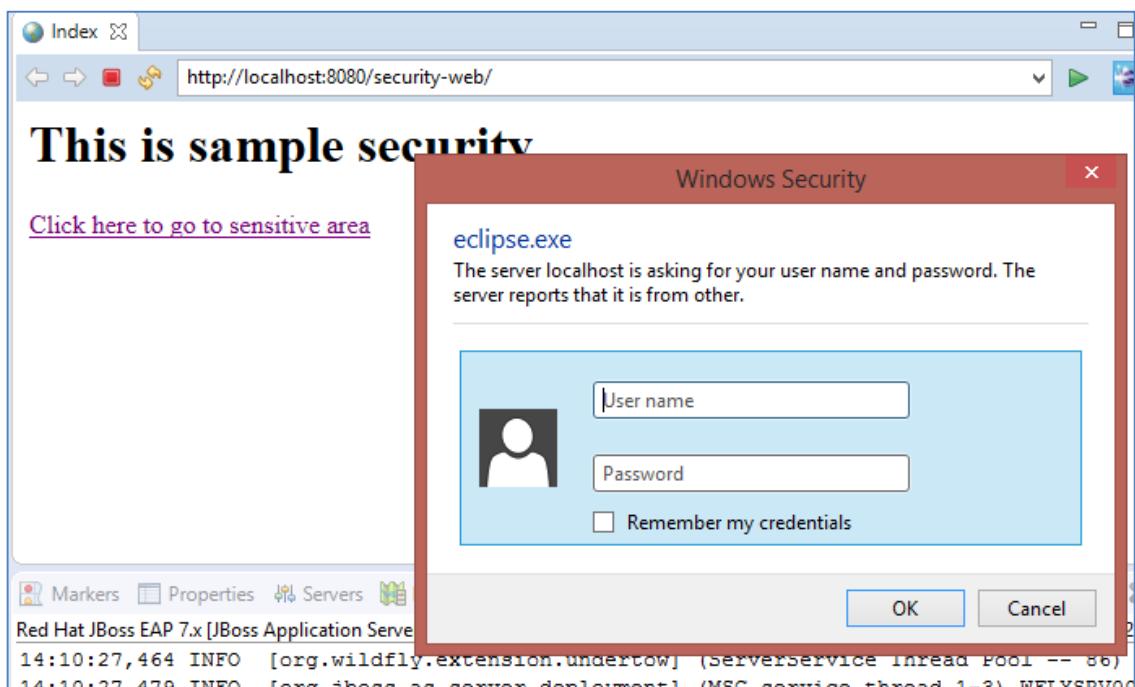
What type of user do you wish to add?
  a) Management User (mgmt-users.properties)
  b) Application User (application-users.properties)
<a>: b

```

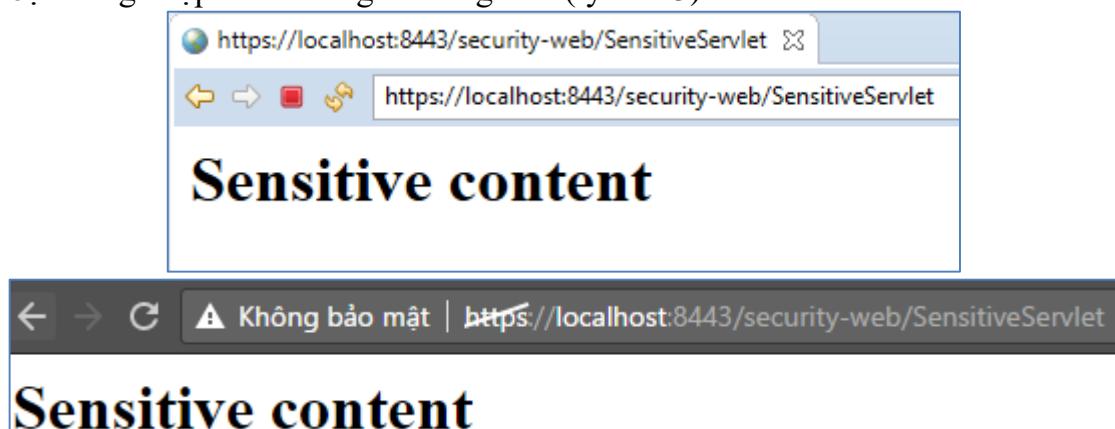
Trong ví dụ này ta thêm 2 user như sau:

- user ty có password 123 thuộc nhóm employee
- user teo có password 123 thuộc nhóm guard
- user det có password 123 thuộc nhóm admin

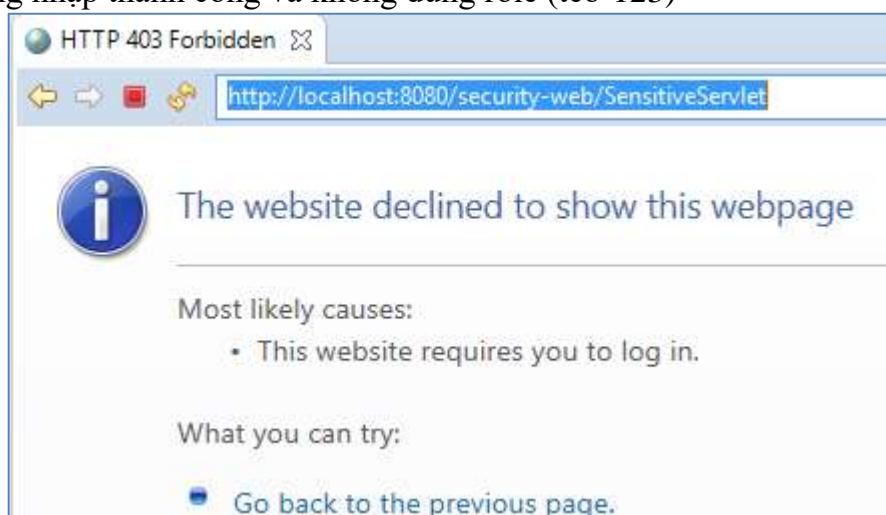
3. Triển khai

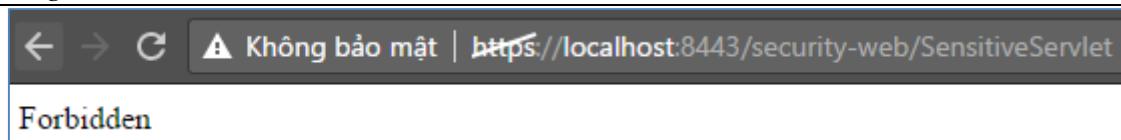


Nếu bạn đăng nhập thành công và đúng role (ty – 123)

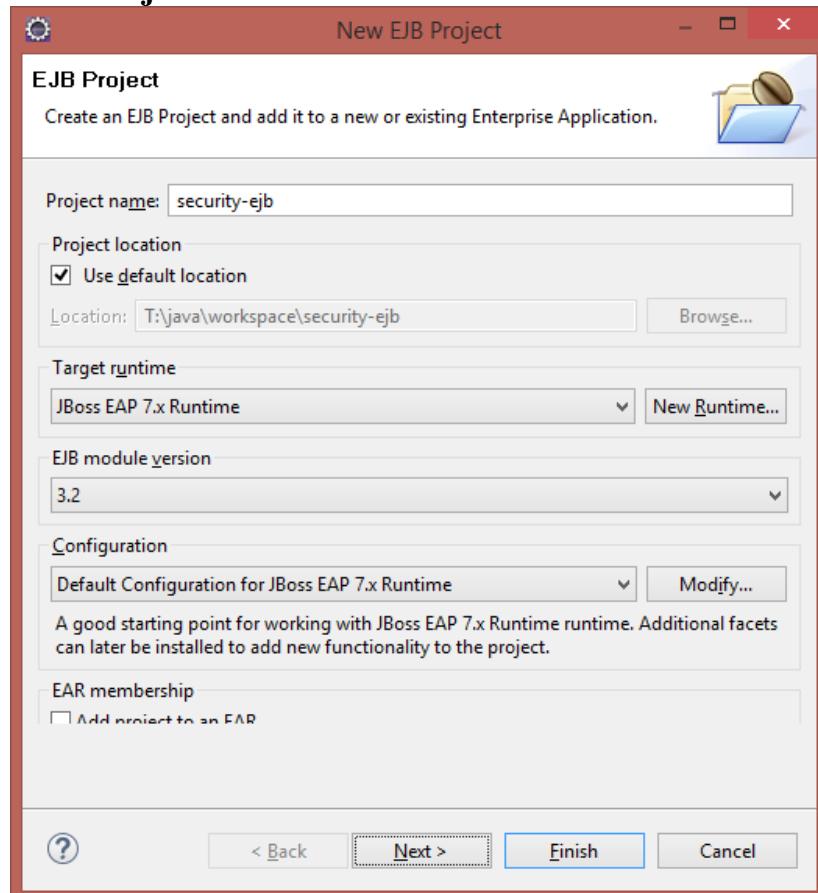


Nếu bạn đăng nhập thành công và không đúng role (teo-123)





Bước 6: Tạo 1 EJB Project



Thêm một session bean có tên SecureEJBsample, nội dung như sau:

```
package vovanhai.wordpress.com;

import javax.annotation.Resource;
import javax.annotation.security.DeclareRoles;
import javax.annotation.security.DenyAll;
import javax.annotation.security.PermitAll;
import javax.annotation.security.RolesAllowed;
import javax.ejb.LocalBean;
import javax.ejb.SessionContext;
import javax.ejb.Stateless;
import org.jboss.security.annotation.SecurityDomain;

@Stateless
@LocalBean
@SecurityDomain("other")
@DeclareRoles({"employee", "guard", "admin"})
public class SecureEJBsample {

    @Resource
    private SessionContext sctx;

    public SecureEJBsample() {
```

```

    }

    @RolesAllowed({"employee", "admin"})
    public String sampleSecureWorksForRoleEmployee(){
        return sctx.getCallerPrincipal()+" : working";
    }

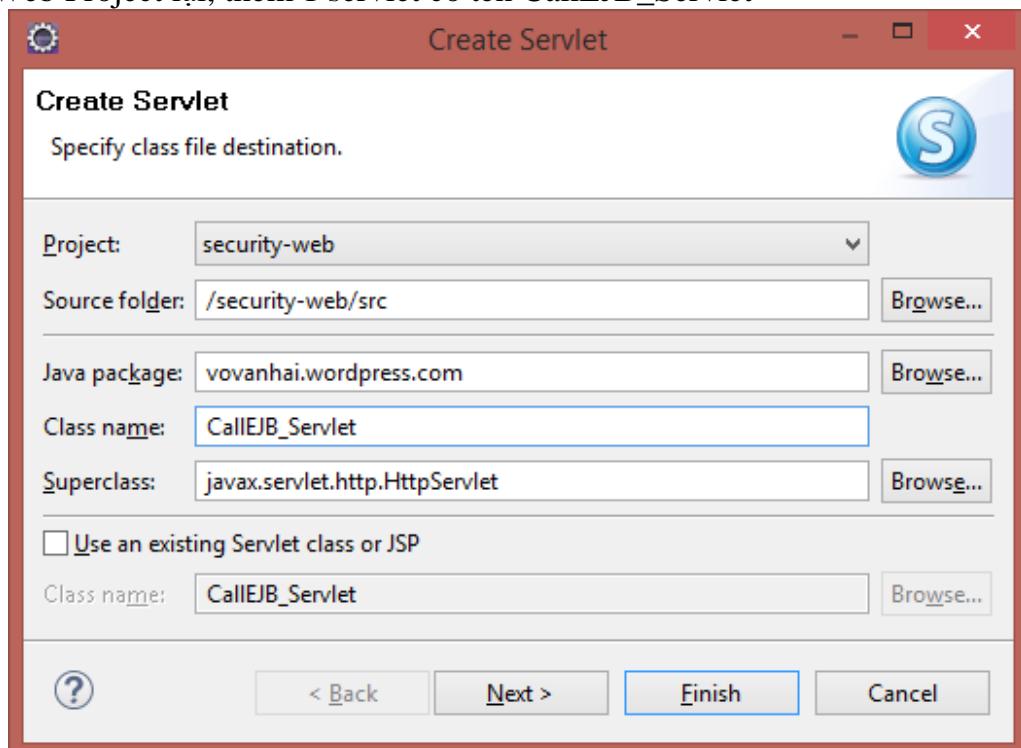
    @RolesAllowed({"guard", "admin"})
    public String sampleSecureWorksForRoleGuard(){
        return sctx.getCallerPrincipal()+" : working";
    }

    @DenyAll
    public void workDenyAll(){
        System.out.println(sctx.getCallerPrincipal()+" : denied all");
    }

    @PermitAll
    public String workPermitAll(){
        return sctx.getCallerPrincipal()+" : allowed all";
    }
}

```

Chỉnh Web Project lại, thêm 1 servlet có tên CallEJB_Servlet



```

package vovanhai.wordpress.com;

import java.io.IOException;
import java.io.PrintWriter;
import javax.ejb.EJB;
import javax.ejb.EJBAccessException;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

@WebServlet("/CallEJB_Servlet")
public class CallEJB_Servlet extends HttpServlet {

```

```

private static final long serialVersionUID = 1L;

@EJB
private SecureEJBsample ejb;

public CallEJB_Servlet() {
    super();
}

protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    PrintWriter out = response.getWriter();
    out.print("<h2>just for role employee, admin</h2>");
    try {
        //for role employee, admin
        String emp = ejb.sampleSecureWorksForRoleEmployee();
        out.println(emp);
    } catch (EJBAccessException e) {
        out.print("Error:"+e.getMessage()+"<br/>");
    }
    out.print("<hr/>");
    out.print("<h2>just for role guard, admin</h2>");
    try {
        //for role guard, admin
        String g = ejb.sampleSecureWorksForRoleGuard();
        out.println(g);
    } catch (EJBAccessException e) {
        out.print("Error:"+e.getMessage());
    }
    out.print("<hr/>");
    out.println("<h2>No-one can access.</h2>");
    try {
        //deny all
        ejb.workDenyAll();
    } catch (EJBAccessException e) {
        out.print("Error:"+e.getMessage());
    }
    out.print("<hr/>");
    out.println("<h2>Everyone can access</h2>");
    //for everyone access
    String pa=ejb.workPermitAll();
    out.println(pa);

    out.close();
}
protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    doGet(request, response);
}
}

```

Trang index.jsp

```

<%@ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8"%>
<html>
<head>
<title>Index</title>
</head>
<body>
    <h1>This is sample security</h1>
    <a href="SensitiveServlet">Click here to go to sensitive area</a><br/>
    <a href="CallEJB_Servlet">Click here to call secure secure ejb</a>
</body>
</html>

```

File web.xml được cấu hình lại

```

<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://xmlns.jcp.org/xml/ns/javaee"
  xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
  http://xmlns.jcp.org/xml/ns/javaee/web-app_3_1.xsd"
  id="WebApp_ID" version="3.1">
  <display-name>security-web</display-name>
  <welcome-file-list>
    <welcome-file>index.html</welcome-file>
    <welcome-file>index.jsp</welcome-file>
  </welcome-file-list>
  <security-constraint>
    <display-name>secure web</display-name>
    <web-resource-collection>
      <web-resource-name>secure test</web-resource-name>
      <url-pattern>/SensitiveServlet</url-pattern>
      <url-pattern>/CallEJB_Servlet</url-pattern>
    </web-resource-collection>
    <auth-constraint>
      <role-name>employee</role-name>
      <role-name>admin</role-name>
      <role-name>guard</role-name>
    </auth-constraint>
    <user-data-constraint>
      <description>transfer data guaranty</description>
      <transport-guarantee>CONFIDENTIAL</transport-guarantee>
    </user-data-constraint>
  </security-constraint>
  <login-config>
    <auth-method>BASIC</auth-method>
    <realm-name>other</realm-name>
  </login-config>
</web-app>

```

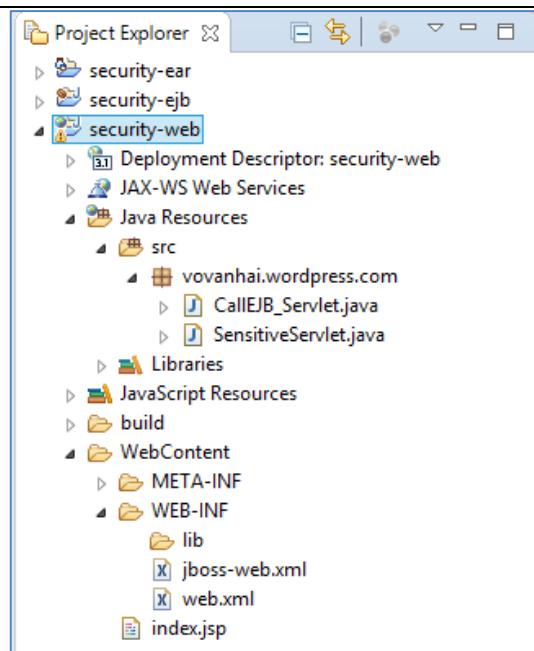
Thêm file jboss-web.xml vào thư mục WEB-INF

```

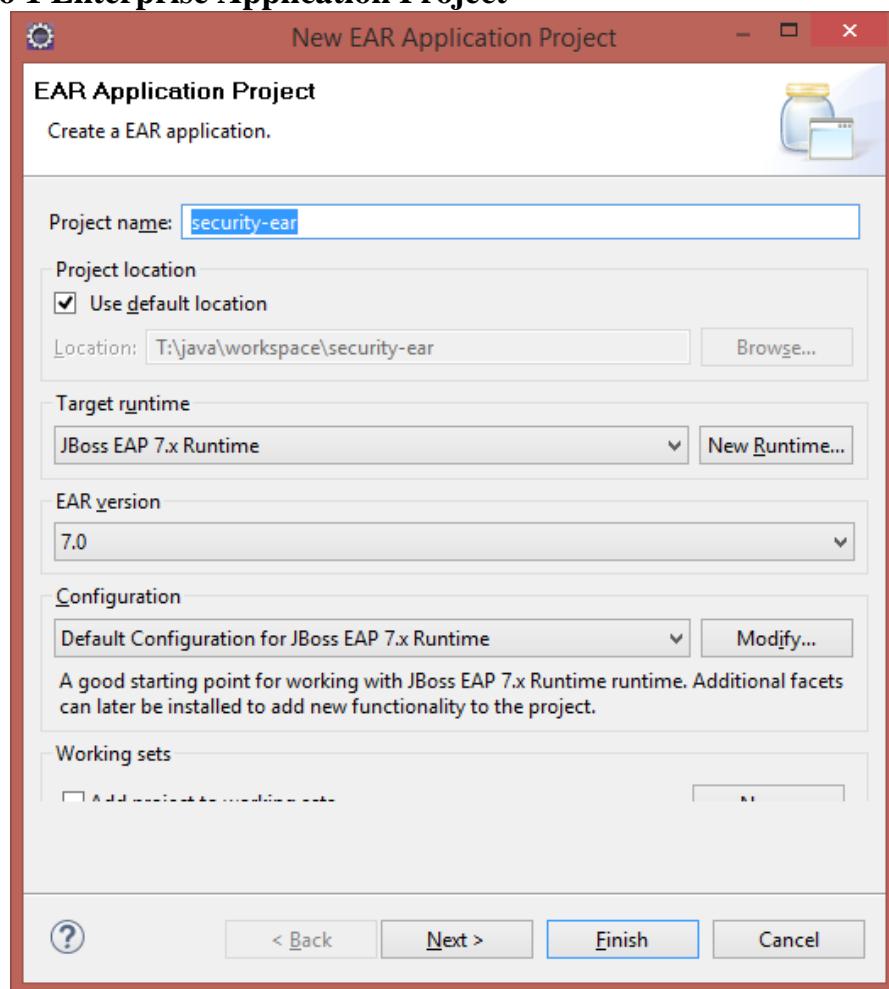
<?xml version="1.0" encoding="UTF-8"?>
<jboss-web>
  <security-domain>other</security-domain>
</jboss-web>

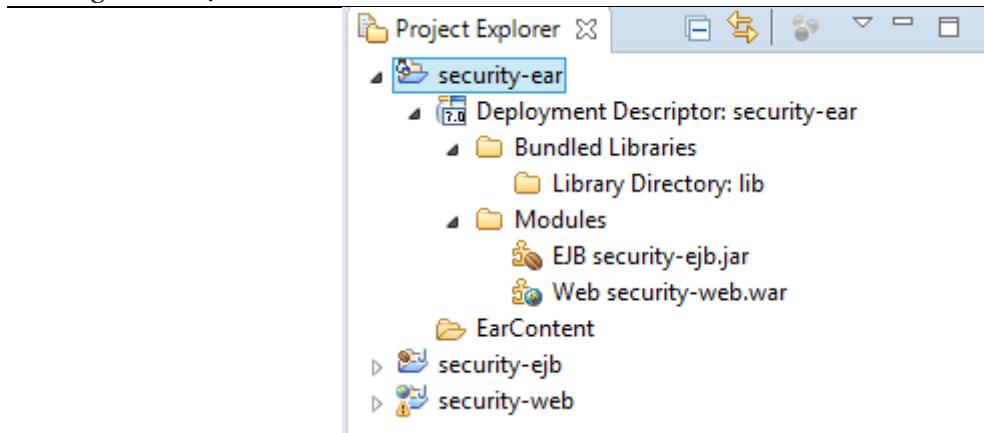
```

Cấu trúc như sau

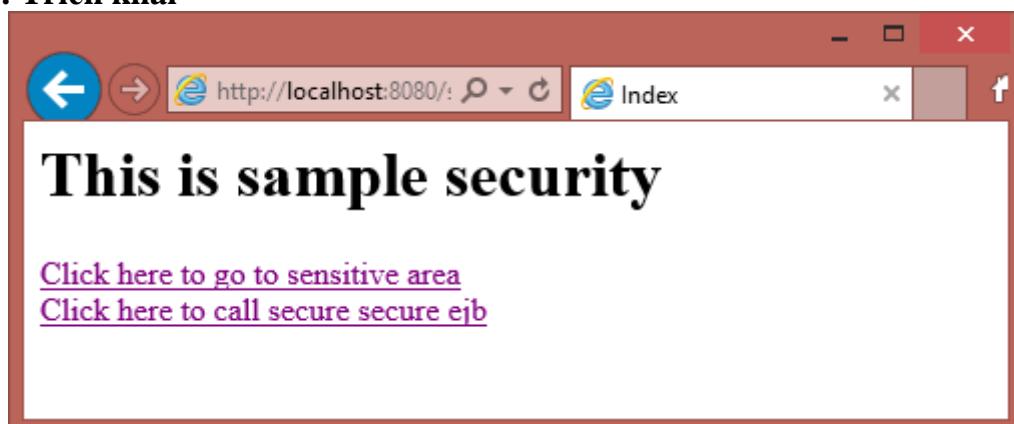


Bước 7: Tạo 1 Enterprise Application Project





Bước 8: Triển khai



Login với role guard<user:teo-psw:123>

just for role employee, admin

Error:WFLYEJB0364: Invocation on method: public java.lang.String vovanhai.wordpress.com.SecureEJBsample.sampleSecureWorksForRoleEmployee() of bean: SecureEJBsample is not allowed

just for role guard, admin

teo: working

No-one can access.

Error:WFLYEJB0364: Invocation on method: public void vovanhai.wordpress.com.SecureEJBsample.workDenyAll() of bean: SecureEJBsample is not allowed

Everyone can access

teo: allowed all

Login với role employee<user:ty-psw:123>

The screenshot shows a web browser window with the URL `https://localhost`. The page content is as follows:

just for role employee, admin

ty: working

just for role guard, admin

Error:WFL YEJB0364: Invocation on method: public java.lang.String vovanhai.wordpress.com.SecureEJBsample.sampleSecureWorksForRoleGuard() of bean: SecureEJBsample is not allowed

No-one can access.

Error:WFL YEJB0364: Invocation on method: public void vovanhai.wordpress.com.SecureEJBsample.workDenyAll() of bean: SecureEJBsample is not allowed

Everyone can access

ty: allowed all

Login với role admin<user:det-psw:123>

The screenshot shows a web browser window with the URL `https://localhost`. The page content is as follows:

just for role employee, admin

det: working

just for role guard, admin

det: working

No-one can access.

Error:WFL YEJB0364: Invocation on method: public void vovanhai.wordpress.com.SecureEJBsample.workDenyAll() of bean: SecureEJBsample is not allowed

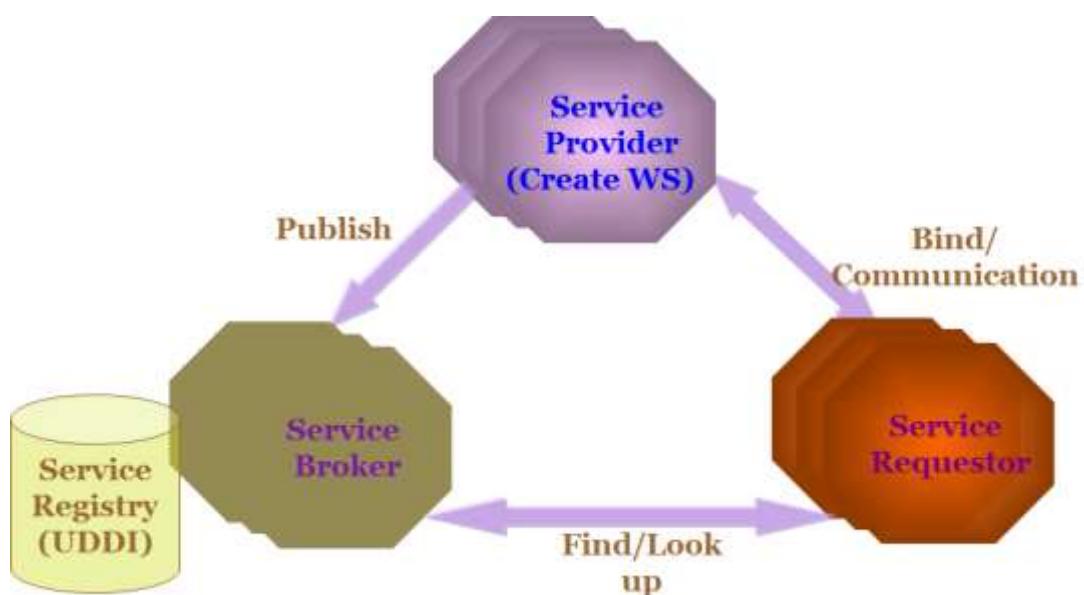
Everyone can access

det: allowed all

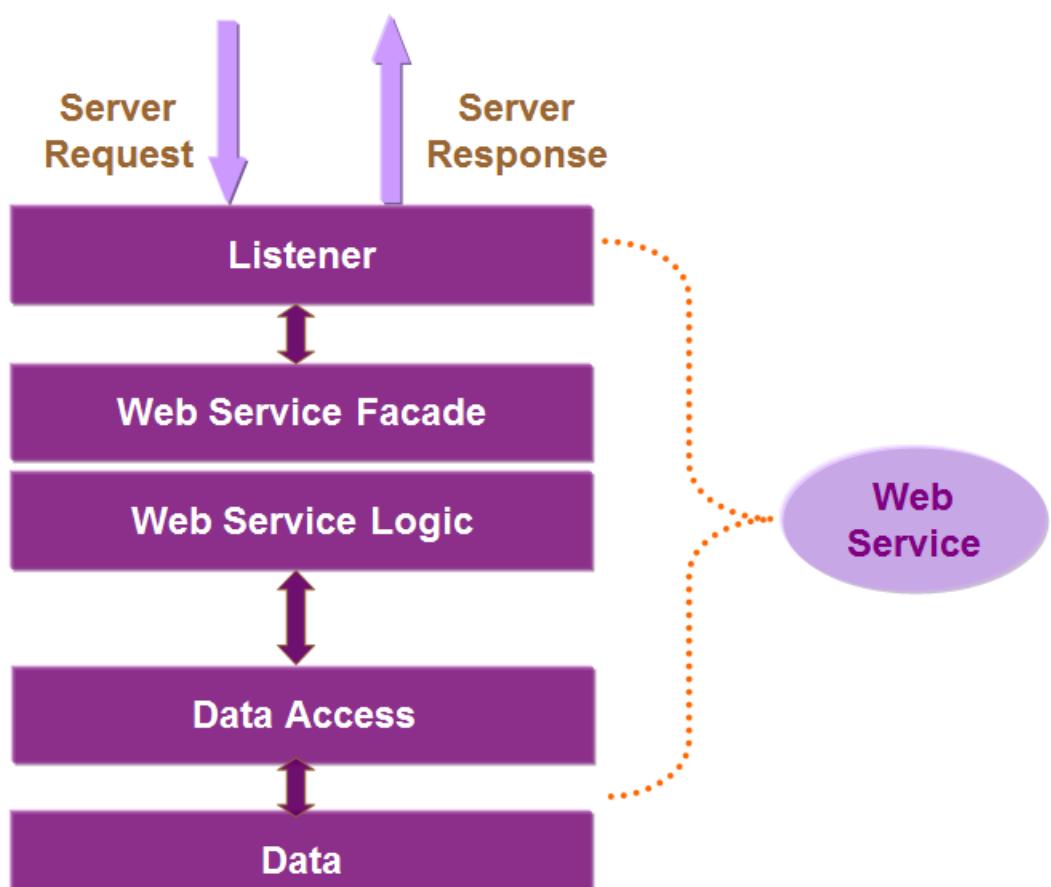
BÀI TẬP

SERVICE-ORIENTED ARCHITECTURE

Components of SOA



SOAP WEBSERVICE

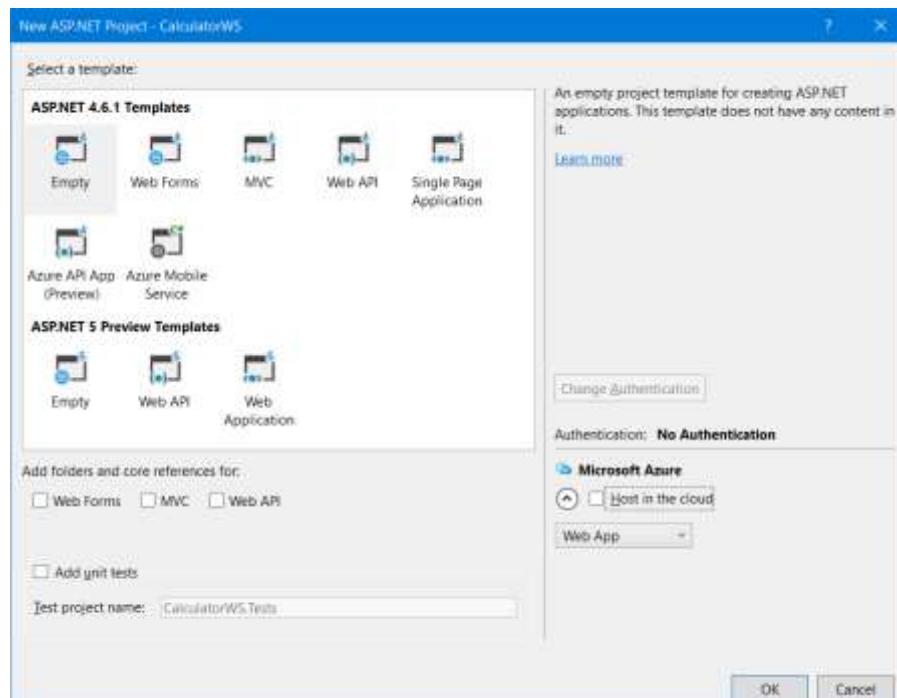
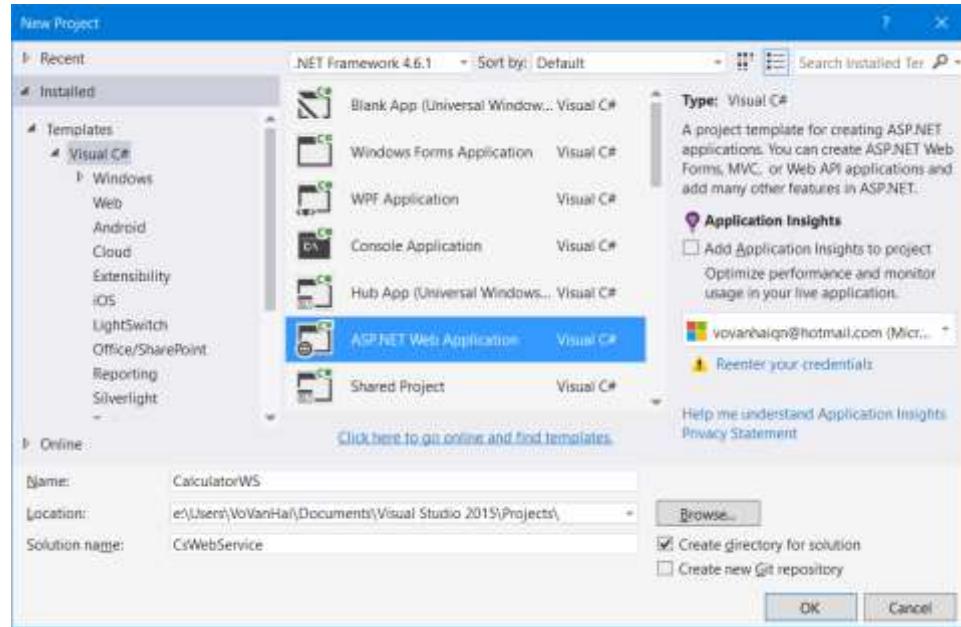


WebService Architecture

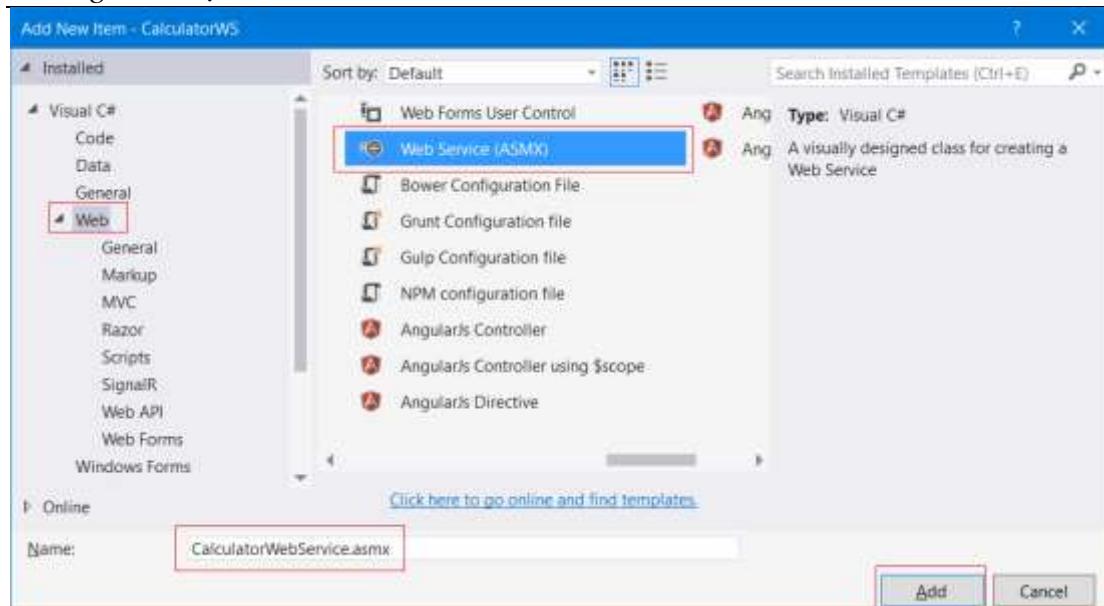
I. SOAP WEB SERVICE VỚI C#

TẠO WEB SERVICE

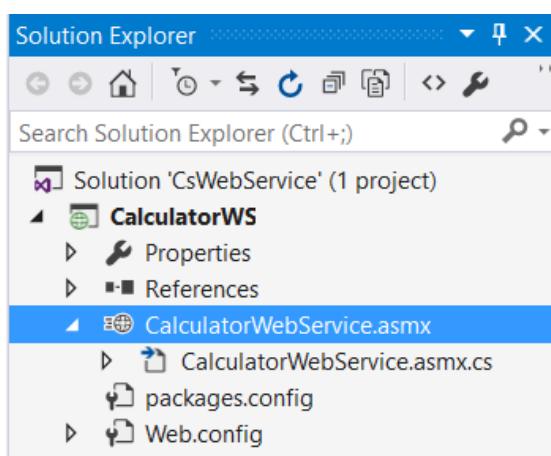
Trong Visual Studio, tạo mới một ASP.NET Web Application. (Dùng IIS Express)



Trong cửa sổ Solution Explorer, nhấn chuột phải lên project, chọn Add → New Item
Tìm chọn mục Web Service (ASMX). Gõ tên CalculatorWebService.asmx



rồi nhấn nút Add. Ta có



VS sẽ sinh cho một lớp có tên `CalculatorWebService` và gắn cho lớp attribute

```
[WebService(Namespace = "http://tempuri.org/")]
```

Ta có thể thay namespace này bằng namespace ta mong muốn

Chú ý mỗi phương thức mà ta muốn xem như một web-method thì ta phải đánh dấu attribute `[WebMethod]`

```

CalculatorWebService.asmx.cs  +
CalculatorWS
1  using System.Web.Services;
2
3  namespace CalculatorWS
4  {
5      /// <summary>
6      /// Summary description for CalculatorWebService
7      /// </summary>
8      [WebService(Namespace = "http://vovanhai.wordpress.com/")]
9      [WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)]
10     [System.ComponentModel.ToolboxItem(false)]
11     // To allow this Web Service to be called from script, using ASP.N
12     // [System.Web.Script.Services.ScriptService]
13     public class CalculatorWebService : System.Web.Services.WebService
14     {
15
16         [WebMethod]
17         public string HelloWorld()
18         {
19             return "Hello World";
20         }
21     }
22 }

```

Tiến hành thêm một số phương thức. Code như sau

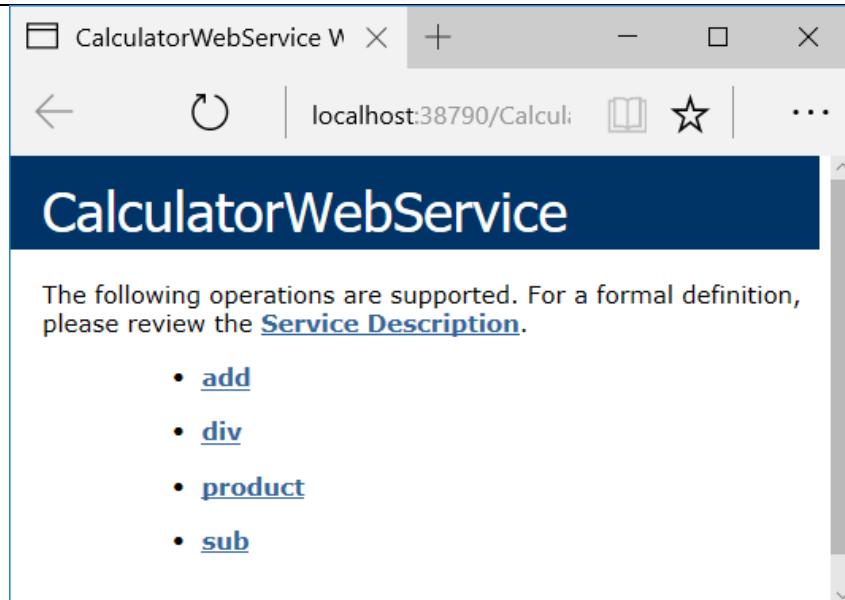
```

using System.Web.Services;

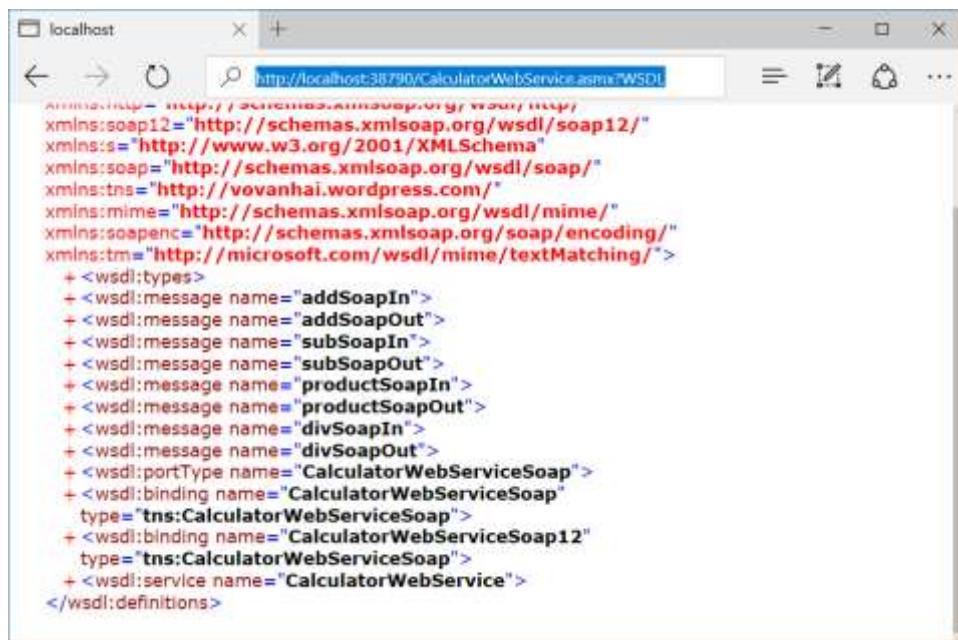
namespace CalculatorWS
{
    /// <summary>
    /// CalculatorWebService cung cấp 4 dịch vụ cho các phép toán cơ bản
    /// </summary>
    [WebService(Namespace = "http://vovanhai.wordpress.com/")]
    [WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)]
    [System.ComponentModel.ToolboxItem(false)]
    [System.Web.Script.Services.ScriptService]
    public class CalculatorWebService : System.Web.Services.WebService
    {
        [WebMethod]
        public double add(double a, double b)
        {
            return a + b;
        }
        [WebMethod]
        public double sub(double a, double b)
        {
            return a - b;
        }
        [WebMethod]
        public double product(double a, double b)
        {
            return a * b;
        }
        [WebMethod]
        public double div(double a, double b)
        {
            if (b == 0) throw new System.Exception("divide by zero");
            return a / b;
        }
    }
}

```

Nhấn F5 để chạy thử trên Microsoft Edge browser. Ta được như sau



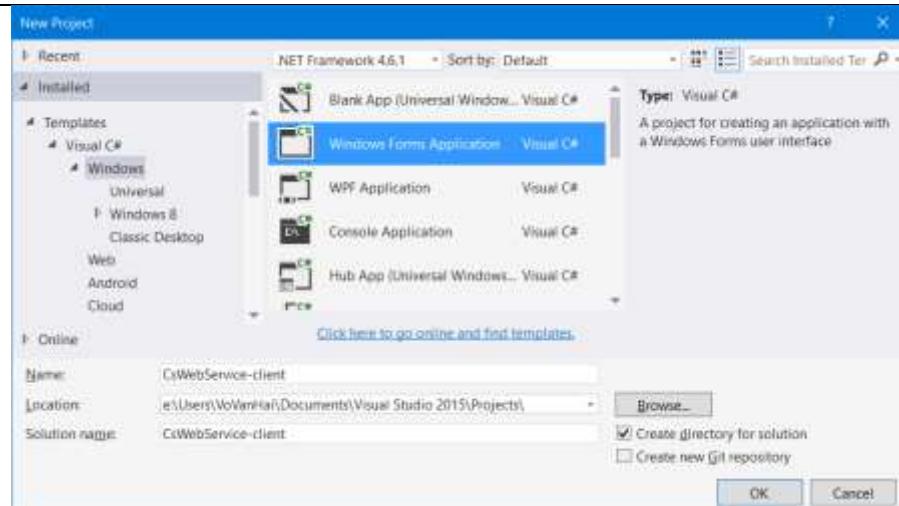
Nhấn Service Description link, ta được



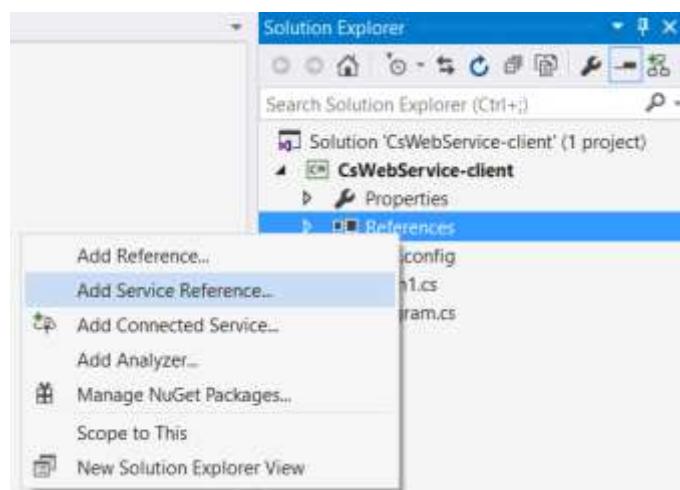
Copy URL này cho việc viết client

CLIENT

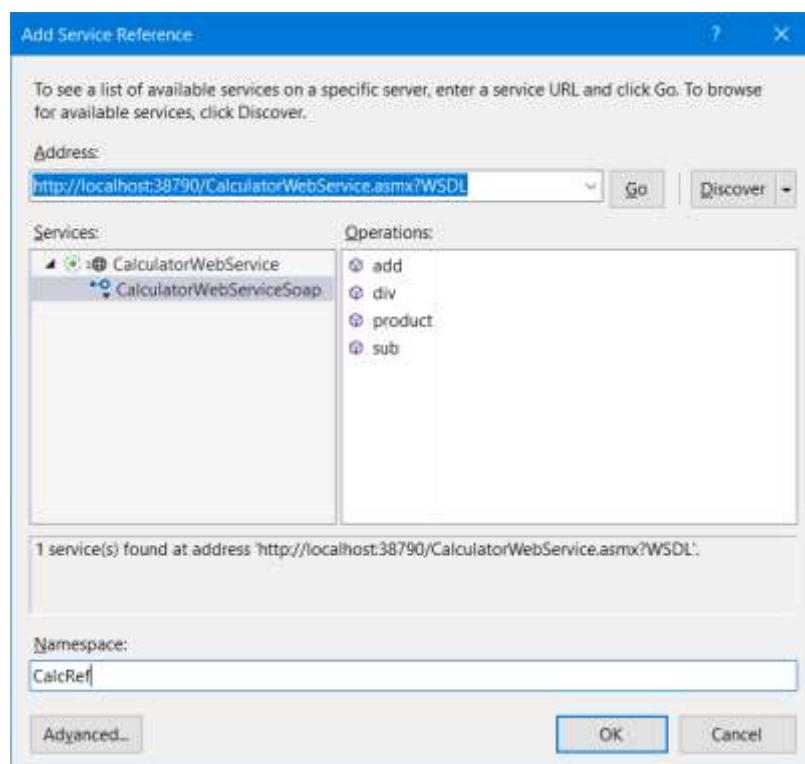
Tạo Windows Form Application có tên CsWebService-client



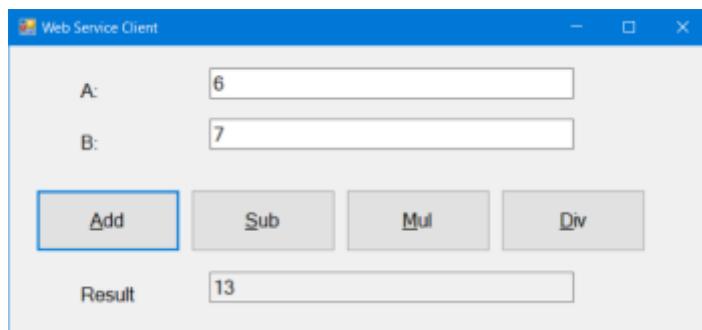
Nhấn phái chuột lên mục References của project, chọn “Add Service Reference..”



Dán uri đã copy ở trên (<http://localhost:38790/CalculatorWebService.asmx?WSDL>) vào



Nhấn nút Go, đặt Namespace lại thành CalcRef như hình trên. Sau đó nhấn OK.



Code cho nút Add như sau. Các nút còn lại code tương tự

```
private void addButton_Click(object sender, EventArgs e)
{
    CsWebService_client.CalcRef.CalculatorWebServiceSoapClient calc = new
CalcRef.CalculatorWebServiceSoapClient();
    double a = double.Parse(aTextBox.Text);
    double b = double.Parse(bTextBox.Text);
    double result=calc.add(a, b);
    resultTextBox.Text = "" + result;
}
```

II. SOAP WEB SERVICE VỚI JAVA**WEBSERVICE WITH ENDPOINT****SERVERT**

1. Tạo service class có tên Calculator với code như sau

```
package vovanhai.wordpress.com;

import javax.jws.WebService;

@WebService(targetNamespace="com.wordpress.vovanhai")
public class Calculator {
    public double add(double a, double b){
        return a + b;
    }
    public double sub(double a, double b){
        return a - b;
    }
    public double product(double a, double b){
        return a * b;
    }
    public double div(double a, double b){
        if (b == 0) throw new RuntimeException("divide by zero");
        return a / b;
    }
}
```

2. Tạo server

```
package vovanhai.wordpress.com;

import javax.xml.ws.Endpoint;

public class VHserver {
    public static void main(String[] args) throws Exception{
        Calculator calc=new Calculator();
        String address="http://localhost:4444/calcws";
        System.out.println("web service bind at "+address);
        Endpoint.publish(address, calc);
    }
}
```

3. Chạy server

Sau khi thực thi server, ta mở trình duyệt lên, gõ địa chỉ sau vào:
<http://localhost:4444/calcws?wsdl>. Ta có kết quả

```

<!-- Generated by JAX-WS RI (http://jax-ws.java.net). RI's version is JAX-WS RI
2.2.9-b130926.1035 svn-revision#5f6196f2b90e9460065a4c2f4e30e065b245e51e. -->
- <definitions name="CalculatorService" targetNamespace="com.wordpress.vovanhai"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:tns="com.wordpress.vovanhai" xmlns="http://schemas.xmlsoap.org/wsdl/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:wsam="http://www.w3.org/2007/05/addressing/metadata"
  xmlns:wsp1_2="http://schemas.xmlsoap.org/ws/2004/09/policy"
  xmlns:wsp="http://www.w3.org/ns/ws-policy" xmlns:wsu="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
  + <types>
  + <message name="add">
  + <message name="addResponse">
  + <message name="div">
  + <message name="divResponse">
  + <message name="sub">
  + <message name="subResponse">
  + <message name="product">
  + <message name="productResponse">
  + <portType name="Calculator">
  + <binding name="CalculatorPortBinding" type="tns:Calculator">
    + <operation name="add">
    + <operation name="div">
    + <operation name="sub">
    + <operation name="product">
  </binding>
</definitions>

```

CLIENT

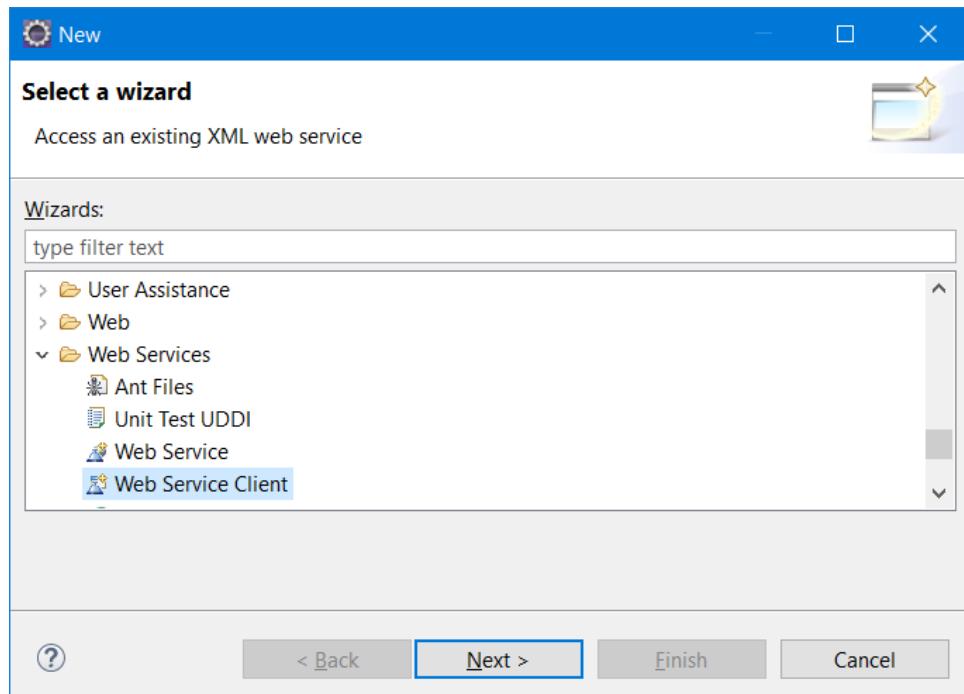
- Có thể chạy command

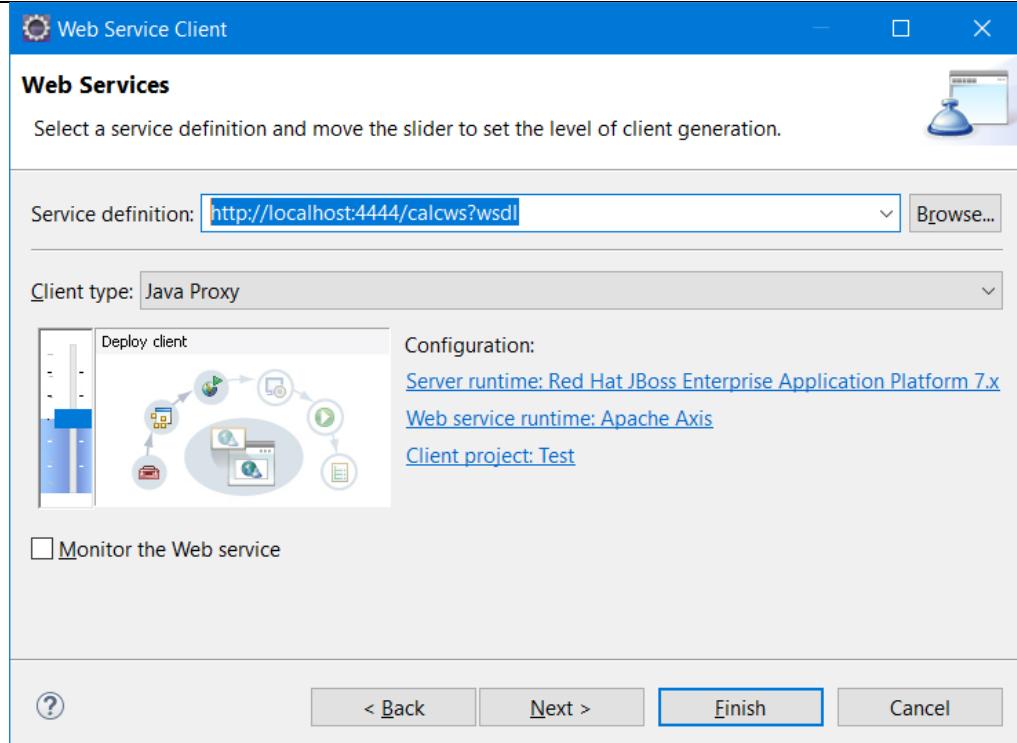
```
wsimport -keep http://localhost:4444/calcws?wsdl
```

để generate stub cho việc triệu gọi webservice. –keep để giữ lại source code java. Sau đó copy các file java vào thư mục client.

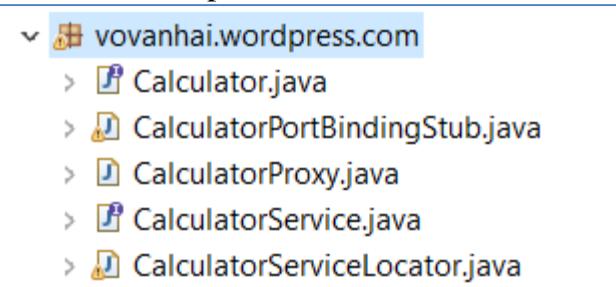
- Hoặc dùng trực tiếp eclipse để generate stub

Nhấn phải chuột lên client project, chọn New->Others. Tìm đến mục Web Services, chọn Web Service Client. Nhấn Next





Copy và dán địa chỉ của wsdl uri vào, nhấn Finish
Eclipse sẽ sinh ra gói vovanhai.wordpress.com chứa các stubs như hình



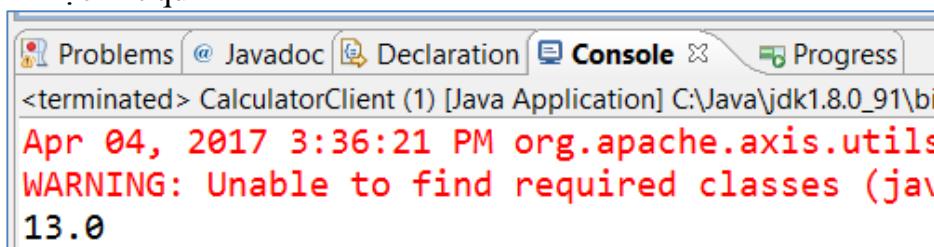
Bây giờ ta thêm lớp client có code như sau

```
package vovanhai.wordpress.com;

import java.rmi.RemoteException;

public class CalculatorClient {
    public static void main(String[] args) throws RemoteException {
        CalculatorProxy proxy=new CalculatorProxy();
        double result=proxy.add(6, 7);
        System.out.println(result);
    }
}
```

Thực thi, ta được kết quả

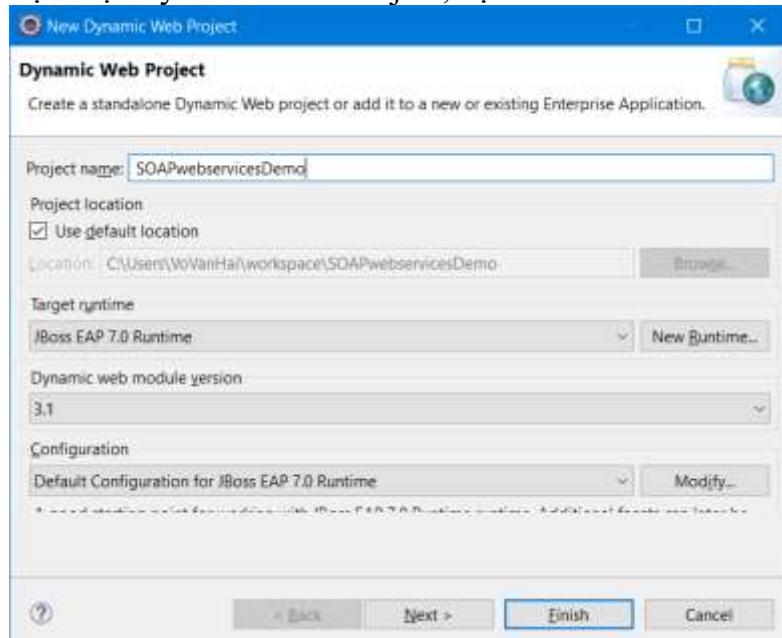


WEBSERVICE ON WEB SERVER

Trong ví dụ này ta hiện thực một webservice cho việc chuyển đổi tiền tệ giữa vnd và usd

SERVER

Tạo một Dynamic Web Project, đặt tên SOAPwebservicesDemo



Chọn Target Runtime là Tomcat hoặc JBossEAP đều được
Nhấn Finish

Tạo một lớp implement dịch vụ chuyển đổi tiền tệ có tên Exchange.

```
package vovanhai.wordpress.com;

import javax.jws.WebMethod;
import javax.jws.WebParam;
import javax.jws.WebService;

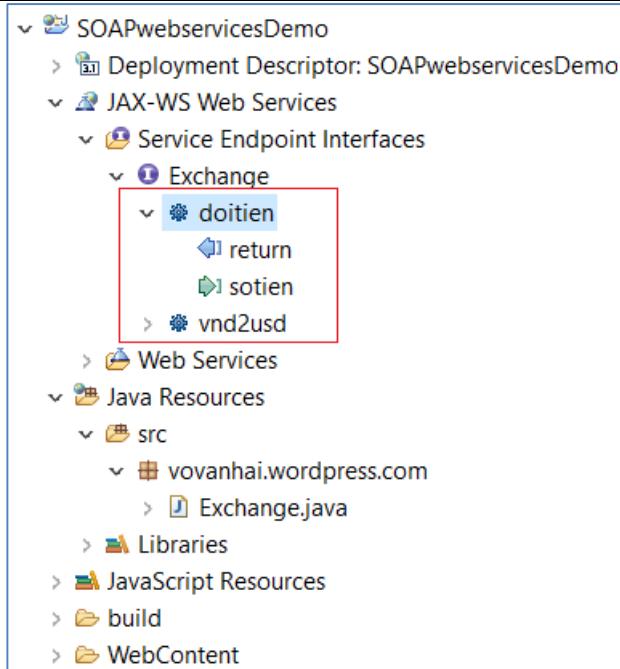
@WebService(targetNamespace="com.wordpress.vovanhai")
public class Exchange {
    private static final double RATE = 23500d;

    @WebMethod(operationName="doitien")
    public double usd2vnd(@WebParam(name="sotien") double amount){
        return amount*RATE;
    }

    public double vnd2usd(double amount){
        return amount*RATE;
    }
}
```

Chú ý:

Chúng ta có thể customize lại các tên của dịch vụ cũng như tên của parameter
Quan sát project của chúng ta bây giờ sẽ như sau:



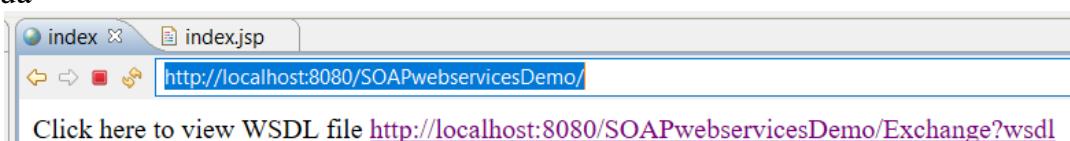
Thêm trang index có nội dung sau

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8"%>
<html>
<head>
<title>index</title>
</head>
<body>
    Click here to view WSDL file
    <a href="http://localhost:8080/SOAPwebserviceDemo/Exchange?wsdl">
        http://localhost:8080/SOAPwebserviceDemo/Exchange?wsdl
    </a>
</body>
</html>
```

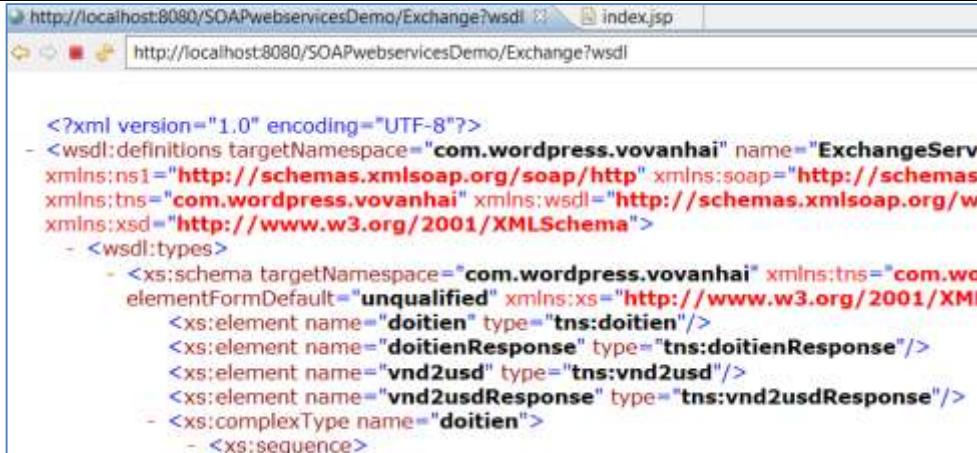
Tiến hành triển khai lên server. Kết quả quan sát trên console như sau

```
address=http://localhost:8080/SOAPwebserviceDemo/Exchange
implementor=vovanhai.wordpress.com.Exchange
serviceName={com.wordpress.vovanhai}ExchangeService
portName={com.wordpress.vovanhai}ExchangePort
annotationWsdlLocation=null
wsdlLocationOverride=null
```

Kết quả



Click link ta được

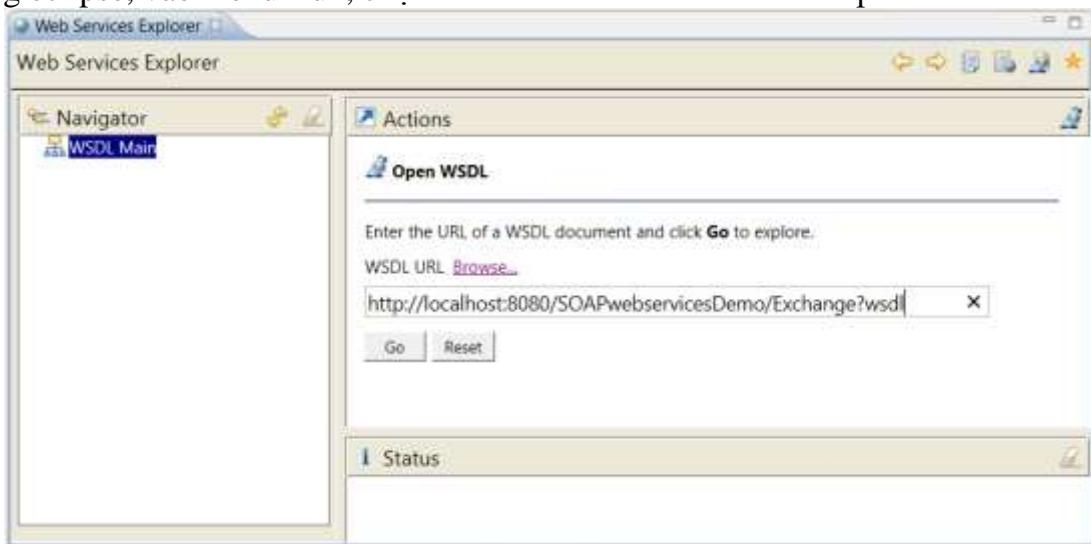


```

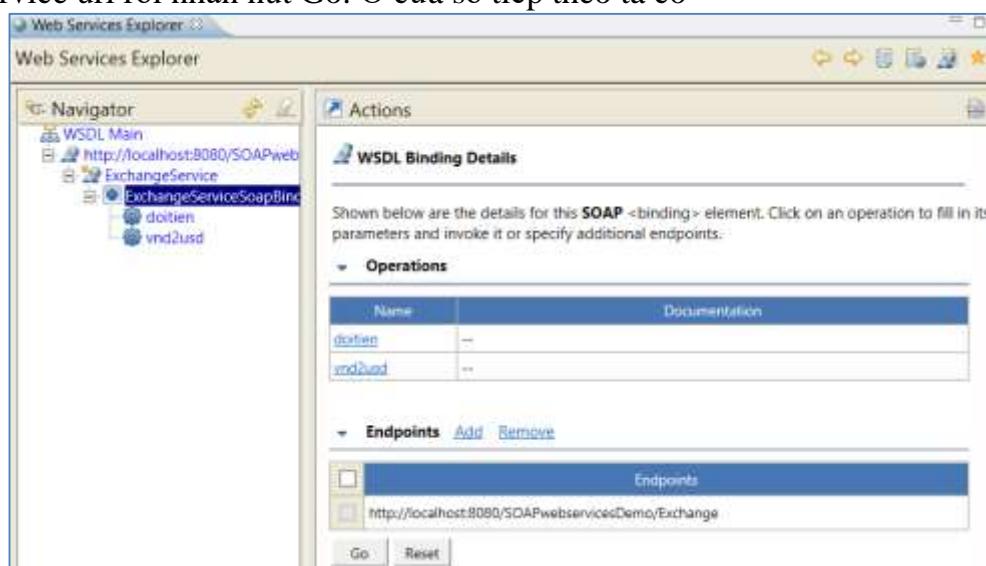
<?xml version="1.0" encoding="UTF-8"?>
- <wsdl:definitions targetNamespace="com.wordpress.vovanhai" name="ExchangeService"
  xmlns:ns1="http://schemas.xmlsoap.org/soap/http" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:tns="com.wordpress.vovanhai" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  - <wsdl:types>
    - <xsd:schema targetNamespace="com.wordpress.vovanhai" xmlns:tns="com.wordpress.vovanhai" elementFormDefault="unqualified" xmlns:xs="http://www.w3.org/2001/XMLSchema">
      <xsd:element name="doitien" type="tns:doitien"/>
      <xsd:element name="doitienResponse" type="tns:doitienResponse"/>
      <xsd:element name="vnd2usd" type="tns:vnd2usd"/>
      <xsd:element name="vnd2usdResponse" type="tns:vnd2usdResponse"/>
    - <xsd:complexType name="doitien">
      - <xsd:sequence>
  
```

CLIENT WITH WEB SERVICES EXPLORER

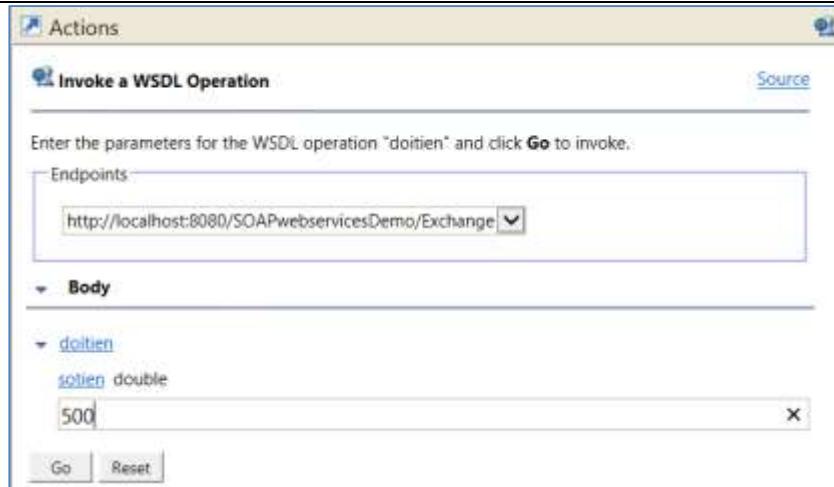
Trong eclipse, vào menu Run, chọn “Launch the Web Services Explorer”



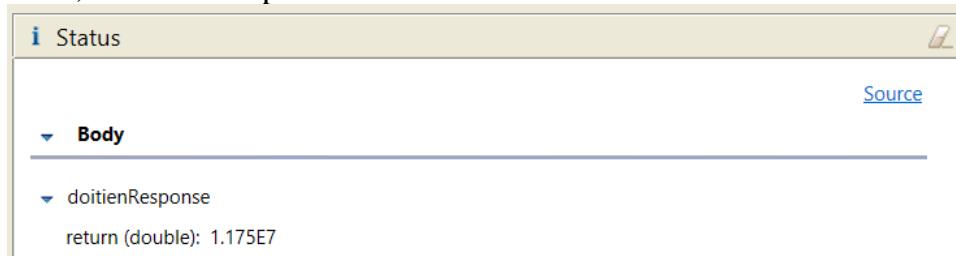
Nhập service url rồi nhấn nút Go. Ở cửa sổ tiếp theo ta có



Nhấn link doitien trong phần Actions mục Operations

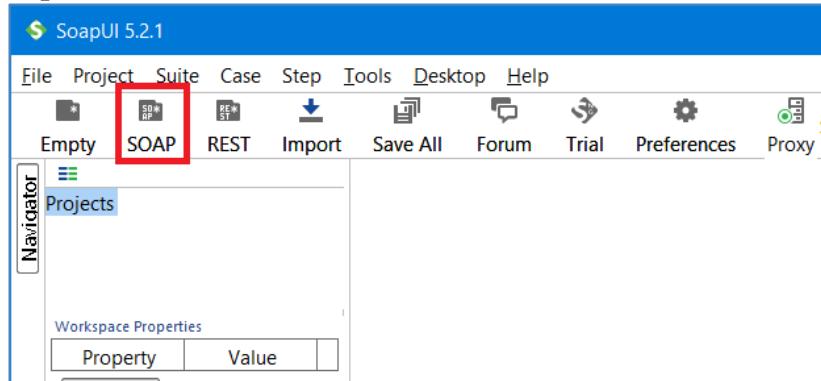


Gõ số tiền 500, nhấn Go. Ở phần Status bên dưới ta có

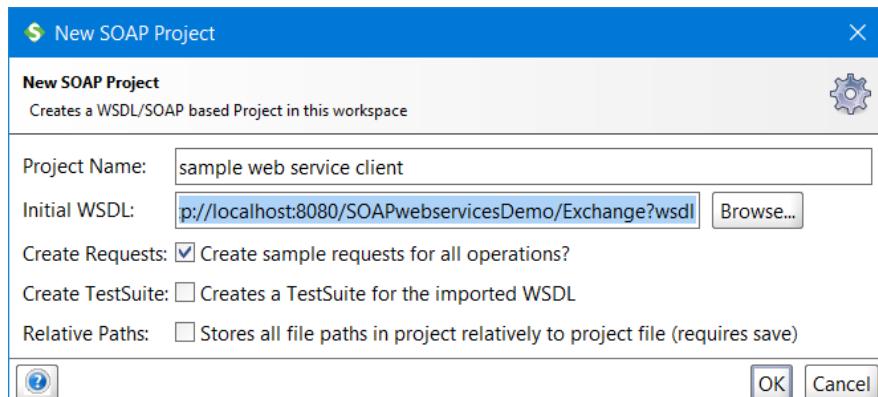


CLIENT WITH SOAP UI

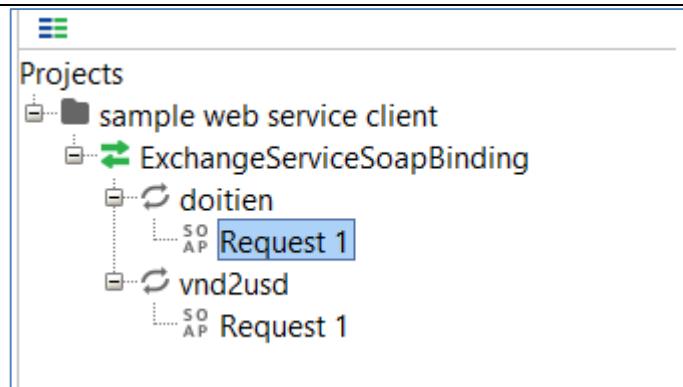
Download tại <http://www.soapui.org>, sau đó giải nén ta được thư mục SOAP UI. Vào bin, chạy file soapui.bat



Nhấn SOAP



Nhấn OK



Double click lên request 1

This screenshot shows the detailed view of 'Request 1' in SoapUI. The top bar indicates the URL is `http://localhost:8080/SOAPwebservicesDemo/Exchange`. The left pane shows the XML for the outgoing request, and the right pane shows the XML for the incoming response. Both panes are labeled 'Raw XML'. Below the panes, there are tabs for Headers, Attachments, SSL Info, WSS, JMS, and JMS Properties. At the bottom, it says 'response time: 23ms (213 bytes)' and has a zoom control '1:1'.

Thay số tiề là 100 sau đó nhấn nút màu xanh góc trên bên trái ta được kết quả như hình trên.

III. SOAP CLIENT WITH CODE

Giả sử ta dùng java Endpoint để sinh một web service như trong mục [WEBSERVICE WITH ENDPOINT](#)

```
package webservice;

import javax.jws.WebParam;
import javax.jws.WebService;

@WebService(targetNamespace="http://vovanhai.wordpress.com")
public class CalculatorWS {
    public int cong(@WebParam(name="a")int a,
                    @WebParam(name="b")int b){
        return a+b;
    }
}

package webservice;

import javax.xml.ws.Endpoint;

public class CaclServer {
    public static void main(String[] args) {
        Endpoint.publish("http://localhost:7777/calcws", new CalculatorWS());
    }
}
```

Code sinh SOAP Message cho việc triệu gọi web service

C#

```
public Stream createMSG()
{
    string soap = @"<?xml version=""1.0"" encoding=""utf-8""?>
    <soapenv:Envelope xmlns:soapenv=""http://schemas.xmlsoap.org/soap/envelope/""
    xmlns:met=""http://met.org.vn"">
        <soapenv:Header>
            <met:number>
                <password>123</password>
                <username>teo</username>
            </met:number>
        </soapenv:Header>
        <soapenv:Body>
            <met:getPrimeNumberUN>
                <number>10</number>
            </met:getPrimeNumberUN>
        </soapenv:Body>
    </soapenv:Envelope>";
    HttpWebRequest req = (HttpWebRequest)WebRequest.Create(
        "http://localhost:6666/primews");
    req.Headers.Add("SOAPAction",
    "\\"http://localhost:6666/getPrimeNumberUN\"");
    req.ContentType = "text/xml; charset=\"utf-8\"";
    req.Accept = "text/xml";
    req.Method = "POST";
    using (Stream stm = req.GetRequestStream())
    {
        using (StreamWriter stmw = new StreamWriter(stm))
        {
            stmw.Write(soap);
        }
    }
    WebResponse response = req.GetResponse();
    Stream responseStream = response.GetResponseStream();
```

```

        return responseStream;
    }
    public string get()
    {
        var st = createMSG();
        StreamReader sr = new StreamReader(st);
        string res = sr.ReadToEnd();
        return res;
    }
}

```

JAVA

```

package webservice;

import java.io.StringWriter;
import java.net.URL;
import javax.xml.namespace.QName;
import javax.xml.soap.MessageFactory;
import javax.xml.soap.SOAPBody;
import javax.xml.soap.SOAPConnection;
import javax.xml.soap.SOAPConnectionFactory;
import javax.xml.soap.SOAPConstants;
import javax.xml.soap.SOAPElement;
import javax.xml.soap.SOAPMessage;
import javax.xml.soap.SOAPPart;
import javax.xml.transform.Source;
import javax.xml.transform.Transformer;
import javax.xml.transform.TransformerFactory;
import javax.xml.transform.stream.StreamResult;

public class CallWsByCode {
    public SOAPMessage createSOAPmessage()throws Exception{
        MessageFactory factory=MessageFactory.newInstance();
        SOAPMessage message=factory.createMessage();
        //xử lý header
        message.getSOAPHeader().detachNode(); //bỏ
        SOAPBody body = message.getSOAPBody();
        //QName qn=new QName("targetURI","Operation","name");
        QName qn=new QName("http://vovanhai.wordpress.com","cong","qn");
        SOAPElement ele = body.addChildElement(qn);
        ele.setEncodingStyle(SOAPConstants.URI_NS_SOAP_ENCODING);
        SOAPElement a = ele.addChildElement("a");
        a.addTextNode("6");
        SOAPElement b = ele.addChildElement("b");
        b.addTextNode("7");

        return message;
    }
    String sendToServer(SOAPMessage msg)throws Exception{
        String xml="";
        URL endpoint=new URL("http://localhost:7777/calcws");
        SOAPConnectionFactory fac=SOAPConnectionFactory.newInstance();
        SOAPConnection con = fac.createConnection();
        SOAPMessage response = con.call(msg, endpoint);
        SOAPPart part = response.getSOAPPart();
        Source src = part.getContent();

        TransformerFactory tf=TransformerFactory.newInstance();
        Transformer trans = tf.newTransformer();
        StringWriter sw=new StringWriter();
        trans.transform(src, new StreamResult(sw));
        xml=sw.toString();
    }
}

```

```

        return xml;
    }
    public static void main(String[] args) throws Exception{
        CallWsByCode x=new CallWsByCode();
        SOAPMessage msg = x.createSOAPmessage();
        msg.writeTo(System.out);
        System.out.println("\n-----\n");
        String xml = x.sendToServer(msg);
        System.out.println(xml);
    }
}

```

Message send được gửi đi có dạng

```

<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Body>
        <qn:cong xmlns:qn="http://vovanhai.wordpress.com" SOAP-
        ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
            <a>6</a>
            <b>7</b>
        </qn:cong>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

Message nhận về có dạng

```

<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
    <S:Body>
        <ns2:congResponse xmlns:ns2="http://vovanhai.wordpress.com">
            <return>13</return>
        </ns2:congResponse>
    </S:Body>
</S:Envelope>

```

IV. SOAP WEB SERVICE TRẢ VỀ JSON FORMAT

Tạo một ASMX Web Service

Tạo một class mô tả cho một nhân viên

```
public class Employee
{
    public int Id { get; set; }
    public string Name { get; set; }
    public int Salary { get; set; }
}
```

Code web service

```
using System.Web.Services;

[WebService(Namespace = "http://vovanhai.wordpress.com/")]
[WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)]
[System.Web.Script.Services.ScriptService]
public class SoapJSONWebService : System.Web.Services.WebService
{
    [WebMethod(Description = "returning the data in JSON")]
    [System.Web.Script.Services.ScriptMethod(ResponseFormat =
        System.Web.Script.Services.ResponseFormat.Json)]
    public void GetEmployessJSON()
    {
        Employee[] emps = new Employee[] {
            new Employee()
            {
                Id=101,
                Name="Teo",
                Salary=5000
            },
            new Employee()
            {
                Id=102,
                Name="Ty",
                Salary=7500
            }
        };
        System.Web.Script.Serialization.JavaScriptSerializer js =
            new System.Web.Script.Serialization.JavaScriptSerializer();
        Context.Response.Write(js.Serialize(emps));
    }
    [WebMethod(Description = "returning the data in XML")]
    public Employee[] GetEmployessXML()
    {
        Employee[] emps = new Employee[] {
            new Employee()
            {
                Id=101,
                Name="teo",
                Salary=5000
            },
            new Employee()
            {
                Id=102,
                Name="ty",
                Salary=7500
            }
        };
        return emps;
    }
}
```

Kết quả

SoapJSONWebService

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

- [GetEmployessJSON](#)
returning the data in JSON
- [GetEmployessXML](#)
returning the data in XML

The screenshot shows two separate browser windows demonstrating the service's functionality.

Top Window: The URL is `localhost/SimpleAuthenticationWS/SoapJSONWebService.asmx/GetEmployessJSON`. The response is a JSON array containing two employee records:

```
[{"Id":101,"Name":"Teo","Salary":5000}, {"Id":102,"Name":"Ty","Salary":7500}]
```

Bottom Window: The URL is `localhost/SimpleAuthenticationWS/SoapJSONWebService.asmx/GetEmployessXML`. The response is an XML document representing the same data:

```
<?xml version="1.0" encoding="UTF-8"?>
<ArrayOfEmployee xmlns="http://vovanhai.wordpress.com/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
    - <Employee>
        <Id>101</Id>
        <Name>teo</Name>
        <Salary>5000</Salary>
    </Employee>
    - <Employee>
        <Id>102</Id>
        <Name>ty</Name>
        <Salary>7500</Salary>
    </Employee>
</ArrayOfEmployee>
```

V. WEB SERVICE SECURITY

SIMPLE AUTHENTICATION

NET WEB SERVICE SIMPLE AUTHENTICATION SECURITY

Cách này đơn giản thêm một header bao gồm một cặp username-password.

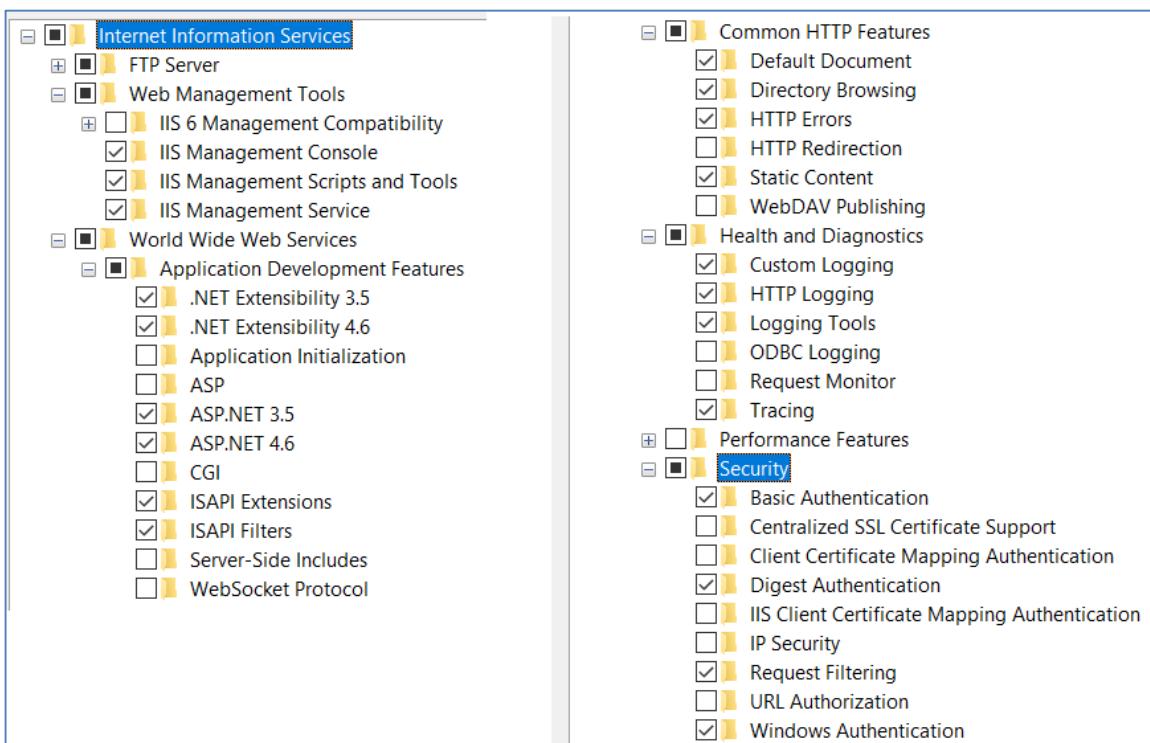
```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
    <soap:Header>
        <AuthenticationHeader xmlns="http://vovanhai.wordpress.com/">
            <userName>string</userName>
            <password>string</password>
        </AuthenticationHeader>
    </soap:Header>
    <soap:Body>
        ...
    </soap:Body>
</soap:Envelope>
```

Ưu điểm: đơn giản

Nhược điểm: security không tốt, ở dạng plain-text

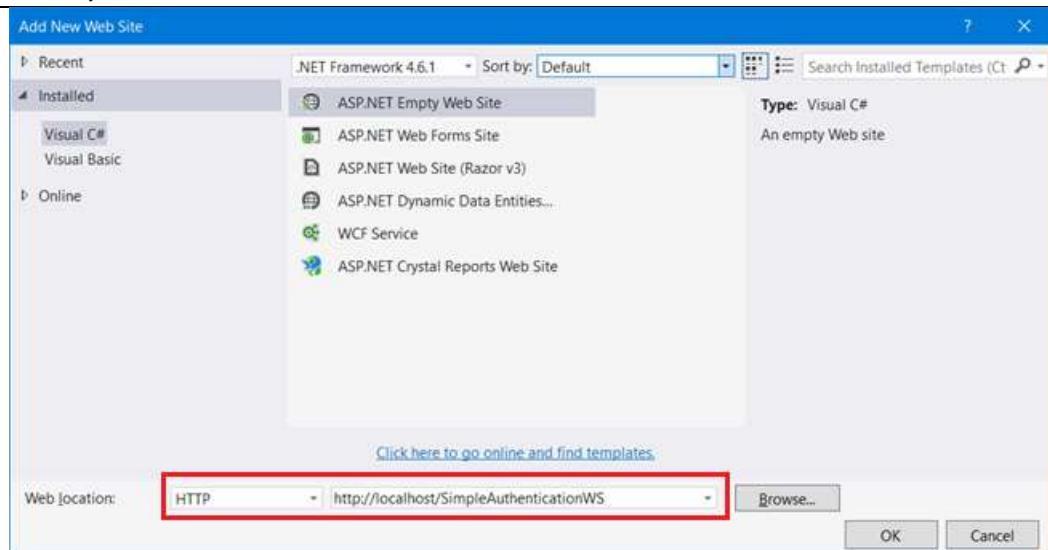
SERVICE

Cấu hình IIS: Nhấn Windows+X, chọn “Programs and Features”. Sau đó chọn “Turn Windows features on or off”



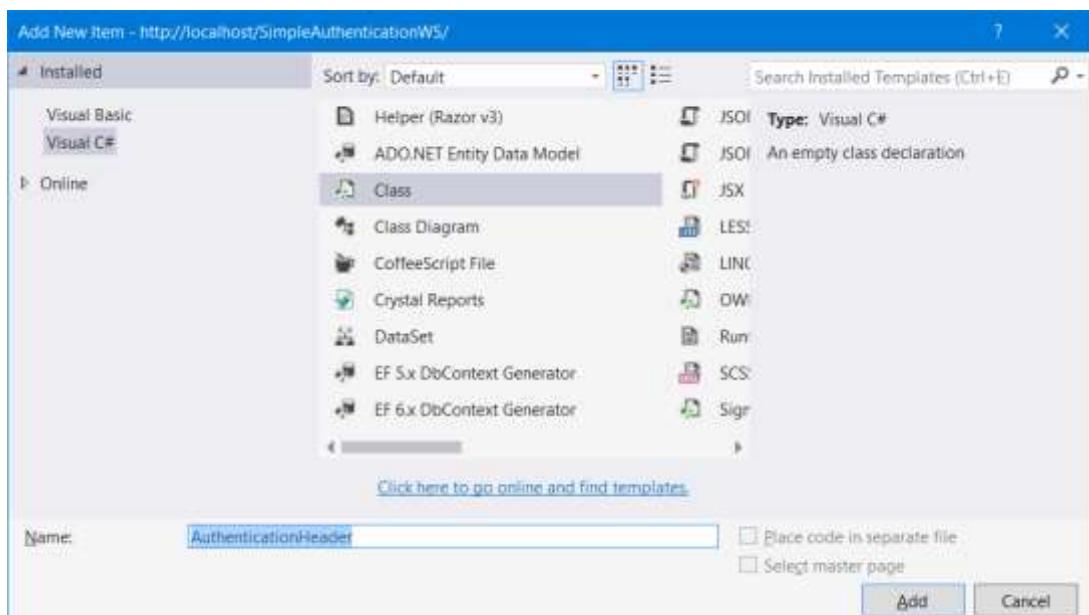
Chạy Visual Studio dưới quyền Administrator

Tạo một web service có tên SimpleAuthenticationWS



Lưu ý: trong ví dụ này ta public lên IIS

Thêm một lớp có tên AuthenticationHeader

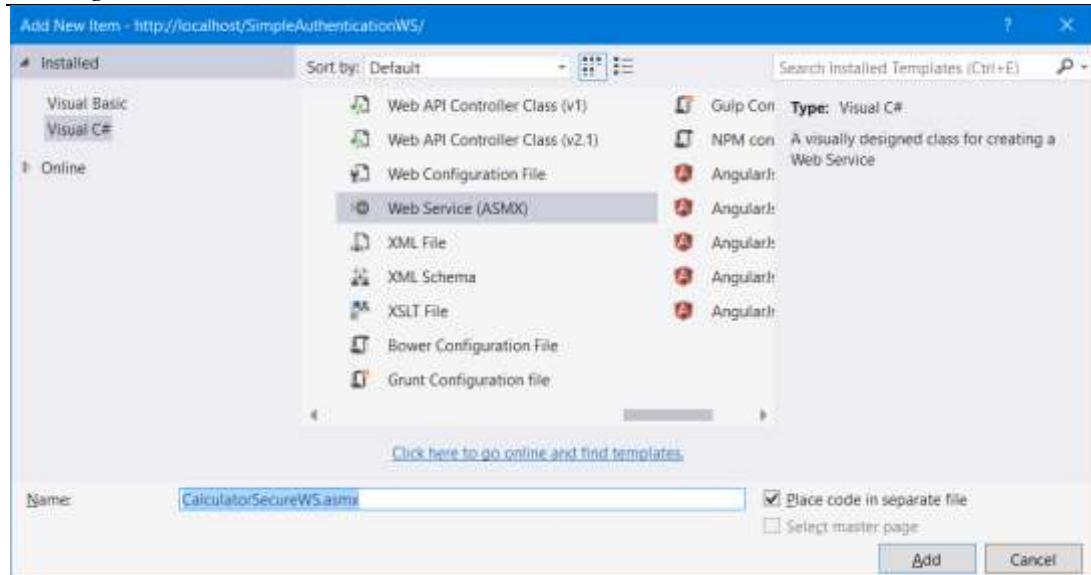


Nội dung của lớp này như sau

```
public class AuthenticationHeader : System.Web.Services.Protocols.SoapHeader
{
    public string userName { get; set; }
    public string password { get; set; }

    public bool IsValid()
    {
        //Write the logic to Check the User Details From DataBase
        //for simple, i use username :teo, and the password is 123
        return this.userName == "teo" && this.password == "123";
    }
}
```

Thêm một Web Service (ASMX) có tên CalculatorSecureWS.asmx



Nội dung của web service này như sau

```

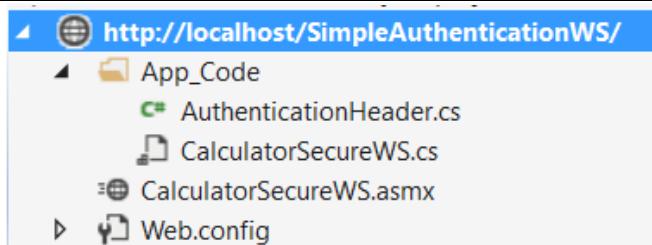
using System;
using System.Web.Services;

[WebService(Namespace = "http://vovanhai.wordpress.com/")]
[WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)]
[System.Web.Script.Services.ScriptService]
public class CalculatorSecureWS : System.Web.Services.WebService
{
    public AuthenticationHeader Authentication;

    [WebMethod	Description = "only for authenticated user"]
    [System.Web.Services.Protocols.SoapHeader("Authentication")]
    public double add(double a, double b)
    {
        if (Authentication == null)
            throw new Exception("Authentication required");
        if (!Authentication.IsValid())
            throw new Exception("Authentication error");
        return a + b;
    }
    [WebMethod	Description = "only for authenticated user"]
    [System.Web.Services.Protocols.SoapHeader("Authentication")]
    public double sub(double a, double b)
    {
        if (Authentication == null)
            throw new Exception("Authentication required");
        if (!Authentication.IsValid())
            throw new Exception("Authentication error");
        return a - b;
    }
    [WebMethod	Description = "everyone can invoke" ]
    public double mul(double a, double b)
    {
        return a * b;
    }
}

```

Cấu trúc project sau khi hoàn tất



Triển khai lên IIS, mở browser duyệt đến <http://localhost/SimpleAuthenticationWS/CalculatorSecureWS.asmx> ta được

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

- [add](#)
only for authenticated user
- [mul](#)
everyone can invoke
- [sub](#)
only for authenticated user

Nhấn mul link ta được

mul

everyone can invoke

Test

To test the operation using the HTTP POST protocol, click the 'Invoke' button.

Parameter	Value
a:	6
b:	7

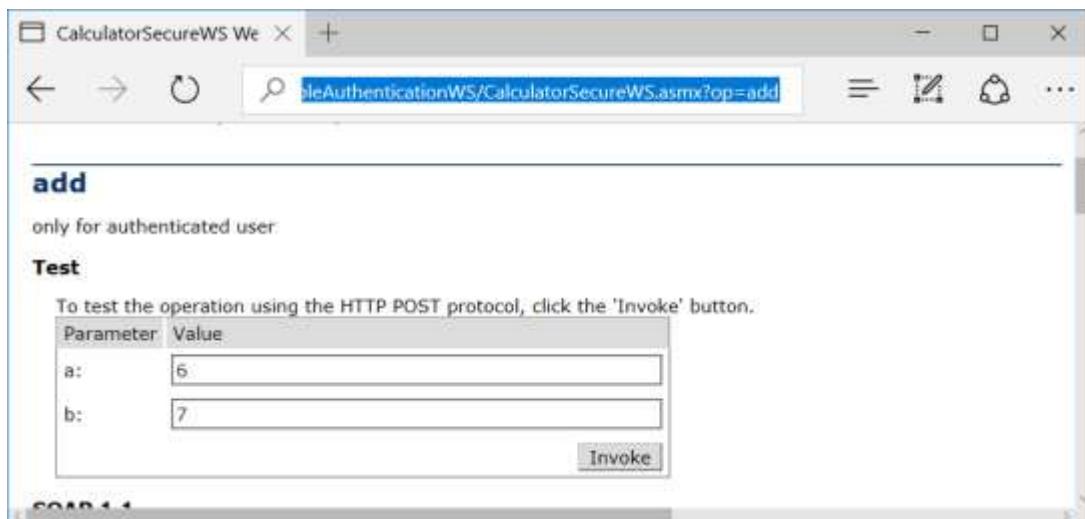
Invoke

Gõ giá trị, nhấn Invoke ta có

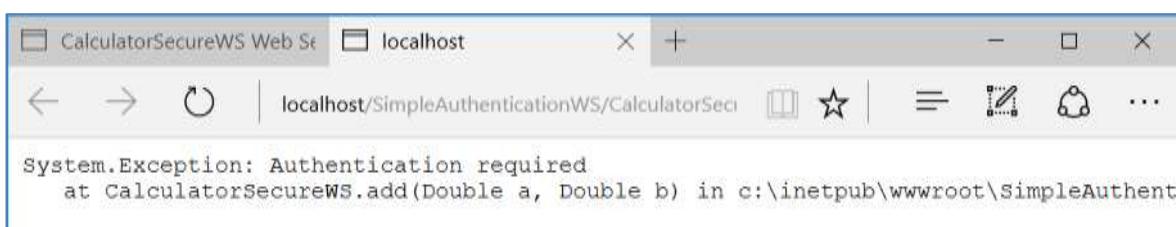
42</double>."/>

```
<?xml version="1.0" encoding="UTF-8"?>
<double xmlns="http://vovanhai.wordpress.com/">42</double>
```

Quay lại, nhấn add link



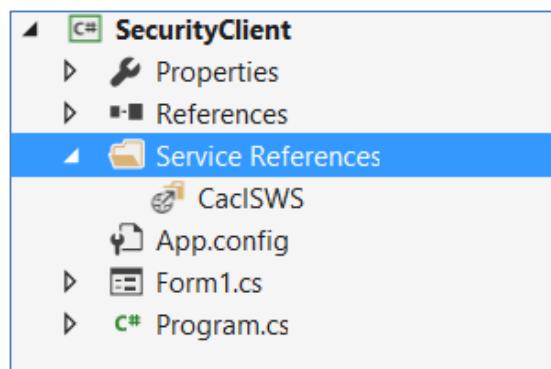
Nhấn Invoke ta có

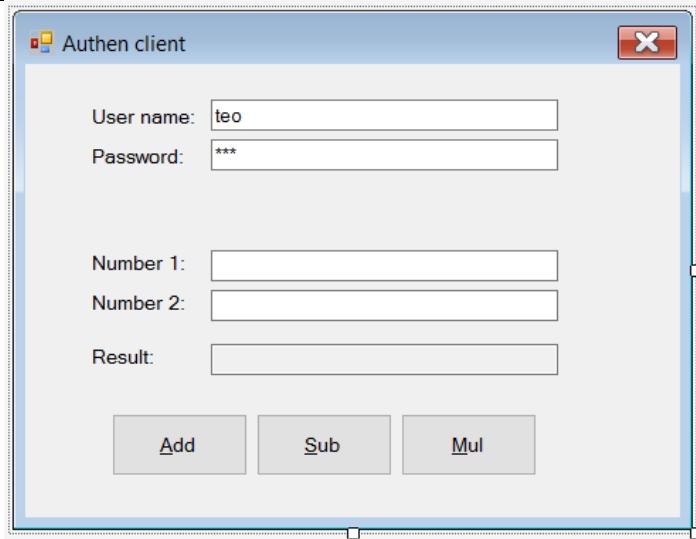


CLIENT

Tạo một Windows Form Application,

Thêm tham chiếu (service references) có tên CalcSWS đến <http://localhost/SimpleAuthenticationWS/CalculatorSecureWS.asmx>





Code cho form

```

using System;
using System.Windows.Forms;
using SecurityClient.CaclsWS;
namespace SecurityClient
{
    public partial class Form1 : Form
    {
        private AuthenticationHeader authen;

        public Form1()
        {
            InitializeComponent();
        }

        private void addButton_Click(object sender, EventArgs e)
        {
            try
            {
                authen = new CaclsWS.AuthenticationHeader();
                authen.userName = usernameTextBox.Text;
                authen.password = passwordTextBox.Text;

                double a = double.Parse(number1TextBox.Text);
                double b = double.Parse(number2TextBox.Text);
                CalculatorSecureWSSoapClient client = new
CaclsWS.CalculatorSecureWSSoapClient();
                double kq = client.add(authen, a, b);
                resultTextBox.Text = "" + kq;
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.Message);
            }
        }

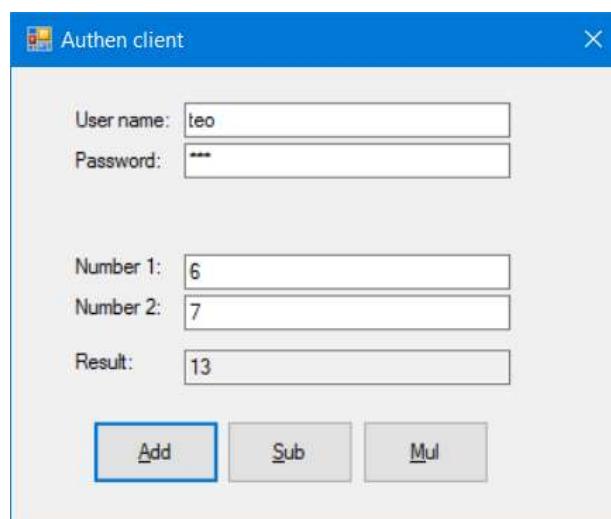
        private void mulButton_Click(object sender, EventArgs e)
        {
            try
            {
                double a = double.Parse(number1TextBox.Text);
                double b = double.Parse(number2TextBox.Text);
                CalculatorSecureWSSoapClient client = new
CaclsWS.CalculatorSecureWSSoapClient();
                double kq = client.mul(a, b);
            }
        }
    }
}

```

```
        resultTextBox.Text = "" + kq;
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}
}
```

Nút Sub tương tự như nút Add

Kết quả



JAVA SIMPLE AUTHENTICATION WEB SERVICE

SERVICE

Tạo lớp AuthenUser để lưu trữ header

```
package vovanhai.wordpress.com;

public class AuthenUser {
    private String username;
    private String password;

    public String getUsername() {
        return username;
    }
    public void setUsername(String username) {
        this.username = username;
    }
    public String getPassword() {
        return password;
    }
    public void setPassword(String password) {
        this.password = password;
    }

    public boolean isAuthenCheck(){
        //kết nối cơ sở dữ liệu để kiểm tra usernam/password
        //trong t/h demo này, dùng us, psw tĩnh
        if (username.equals("teo") && password.equals("123"))
            return true;
        return false;
    }
}
```

Tạo một web service với code sau (chạy trên Endpoint hay WebServer đều được)

Web service này gồm một service kiểm tra một số có nguyên tố hay không và một service trả về một danh sách các số nguyên tố nhỏ hơn hoặc bằng một số cho trước

```
package vovanhai.wordpress.com;

import java.util.ArrayList;
import java.util.List;
import javax.jws.WebMethod;
import javax.jws.WebParam;
import javax.jws.WebService;

@WebService(targetNamespace="http://vovanhai.wordpress.com")
public class PrimeWS {
    public boolean isPrimeNumber(@WebParam(name="number")int number)
    {
        for (int i = 2; i <= number / 2; i++){
            if (number % i == 0)
                return false;
        }
        return true;
    }
    @WebMethod
    public List<Integer>getPrimeNumberUN(
        @WebParam(name="number",header=true) AuthenUser auth,
        @WebParam(name="number")int n
    ){
        if(auth==null)
            throw new RuntimeException("Authentication required");
    }
}
```

```

        if(!auth.isAuthenCheck())
            throw new RuntimeException("Invalid username or password");

        List<Integer> lst = new ArrayList<>();
        for (int i = 2; i <= n; i++){
            if (isPrimeNumber(i))
                lst.add(i);
        }
        return lst;
    }
}

```

header=true để đảm bảo nó được đưa vào header của soap request

CLIENT

- Code với eclipse sau khi dùng tool Web Service Client

```

public class Client {
    public static void main(String[] args) throws Exception{
        PrimeWSProxy proxy=new PrimeWSProxy();
        PrimeWS p = proxy.getPrimeWS();
        boolean kq = p.isPrimeNumber(7);
        System.out.println(kq);
        int[] lst = p.getPrimeNumberUN(20, "123", "teo");
        System.out.println(lst);
    }
}

```

- Code triệu gọi, không dùng tool Web Service Client

```

package webservice;
import java.io.StringWriter;
import java.net.URL;
import javax.xml.namespace.QName;
import javax.xml.soap.MessageFactory;
import javax.xml.soap.SOAPBody;
import javax.xml.soap.SOAPConnection;
import javax.xml.soap.SOAPConnectionFactory;
import javax.xml.soap.SOAPConstants;
import javax.xml.soap.SOAPElement;
import javax.xml.soap.SOAPHeader;
import javax.xml.soap.SOAPMessage;
import javax.xml.soap.SOAPPart;
import javax.xml.transform.Source;
import javax.xml.transform.Transformer;
import javax.xml.transform.TransformerFactory;
import javax.xml.transform.stream.StreamResult;

public class CallWsWithHeaderByCode {
    public SOAPMessage createSOAPmessage()throws Exception{
        MessageFactory factory=MessageFactory.newInstance();
        SOAPMessage message=factory.createMessage();
        //xử lý header
        SOAPHeader header = message.getSOAPHeader();
        QName qname=new QName("http://vovanhai.wordpress.com","number","vov");
        SOAPElement hele = header.addChildElement(qname);
        hele.addChildElement("username").addTextNode("teo");
        hele.addChildElement("password").addTextNode("123");
        //body
        SOAPBody body = message.getSOAPBody();
        //QName qn=new QName("targetNamespace","Operation","name");
        QName qn=new QName("http://vovanhai.wordpress.com","getPrimeNumberUN","prime");
        SOAPElement ele = body.addChildElement(qn);
    }
}

```

```
ele.setEncodingStyle(SOAPConstants.URI_NS_SOAP_ENCODING);
SOAPElement a = ele.addChildElement("number");
a.addTextNode("100");
return message;
}
String sendToServer(SOAPMessage msg) throws Exception{
String xml="";
URL endpoint=new URL("http://localhost:6666/primews");
SOAPConnectionFactory fac=SOAPConnectionFactory.newInstance();
SOAPConnection con = fac.createConnection();
SOAPMessage response = con.call(msg, endpoint);
SOAPPart part = response.getSOAPPart();
Source src = part.getContent();

TransformerFactory tf=TransformerFactory.newInstance();
Transformer trans = tf.newTransformer();
trans.setOutputProperty(javax.xml.transform.OutputKeys.INDENT, "yes");
StringWriter sw=new StringWriter();
trans.transform(src, new StreamResult(sw));
xml=sw.toString();
return xml;
}
public static void main(String[] args) throws Exception{
CallWsWithHeaderByCode x=new CallWsWithHeaderByCode();
SOAPMessage msg = x.createSOAPmessage();
msg.writeTo(System.out);
System.out.println("\n-----\n");
String xml = x.sendToServer(msg);
System.out.println(xml);
}
}
```

VI. WSE

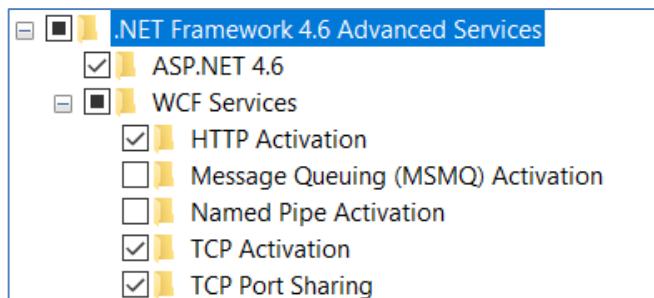
RESOURCE-ORIENTED ARCHITECTURE

Một số Code return của REST

No.	HTTP Code & Description
1	200 OK – shows success.
2	201 CREATED – when a resource is successfully created using POST or PUT request. Returns link to the newly created resource using the location header.
3	204 NO CONTENT – when response body is empty. For example, a DELETE request.
4	304 NOT MODIFIED – used to reduce network bandwidth usage in case of conditional GET requests. Response body should be empty. Headers should have date, location, etc.
5	400 BAD REQUEST – states that an invalid input is provided. For example, validation error, missing data.
6	401 UNAUTHORIZED – states that user is using invalid or wrong authentication token.
7	403 FORBIDDEN – states that the user is not having access to the method being used. For example, Delete access without admin rights.
8	404 NOT FOUND – states that the method is not available.
9	409 CONFLICT – states conflict situation while executing the method. For example, adding duplicate entry.
10	500 INTERNAL SERVER ERROR – states that the server has thrown some exception while executing the method.

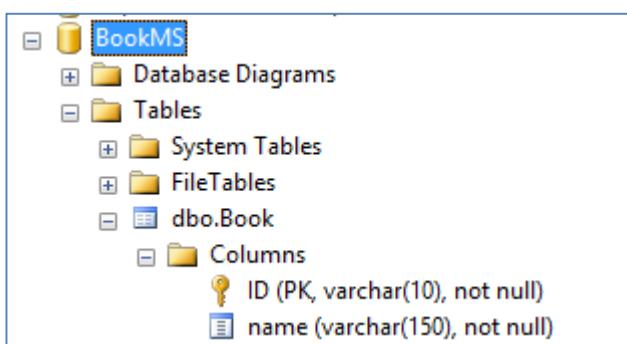
I. WINDOWS COMMUNICATION FOUNDATION (WCF)

CẤU HÌNH IIS



TẠO WCF REST

Tạo cơ sở dữ liệu có tên BookMS và bảng Book có cấu trúc như sau

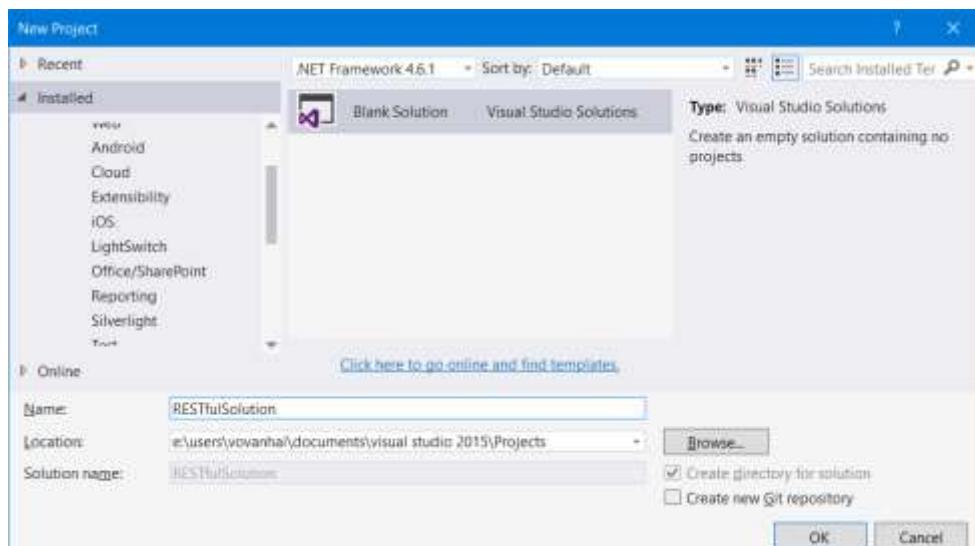


Nhập vài mẫu tin để test

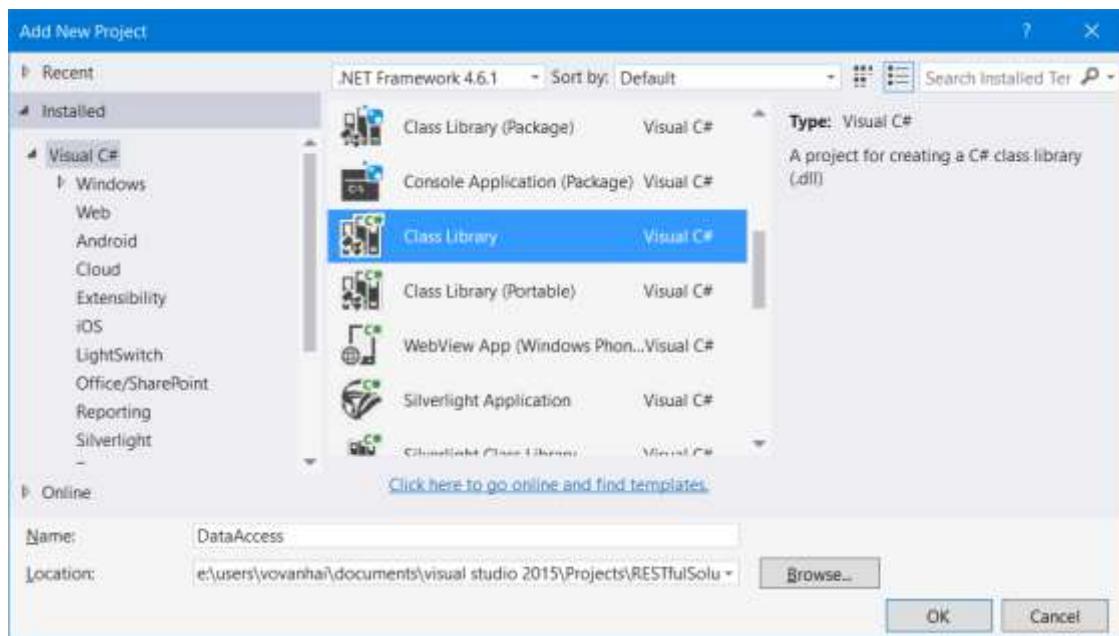
HBMNL.BookMS - dbo.Book		
	ID	name
▶	1001	lập trình java
	1002	C# Nâng cao

Mở Visual Studio với quyền Administrator

Tạo một Blank Solution có tên RESTfulSolution

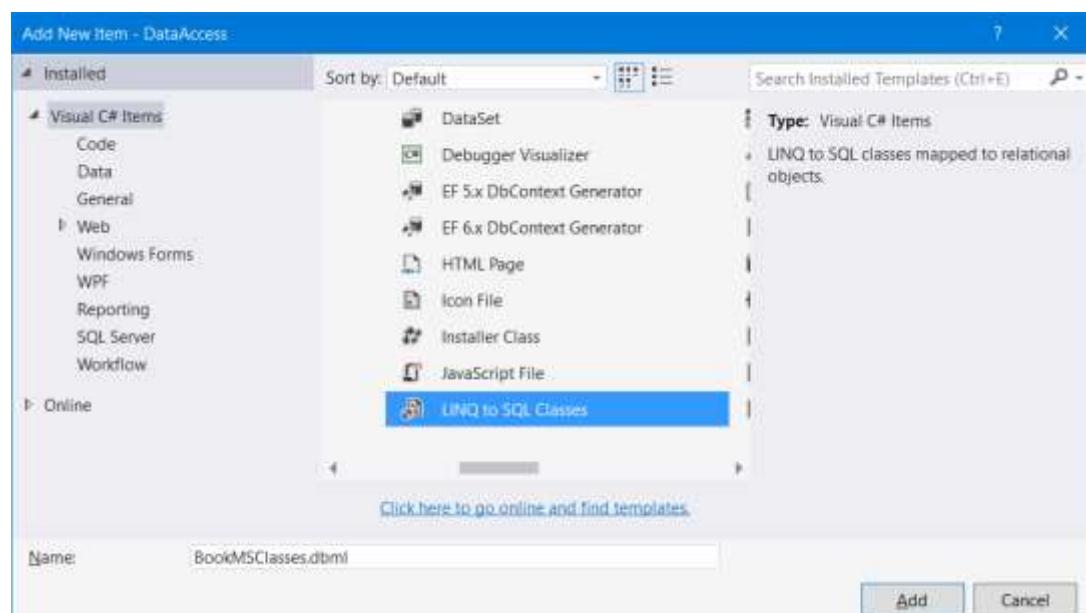


Nhảm phải chuột lên Solution, thêm một ClassLibrary có tên DataAccess

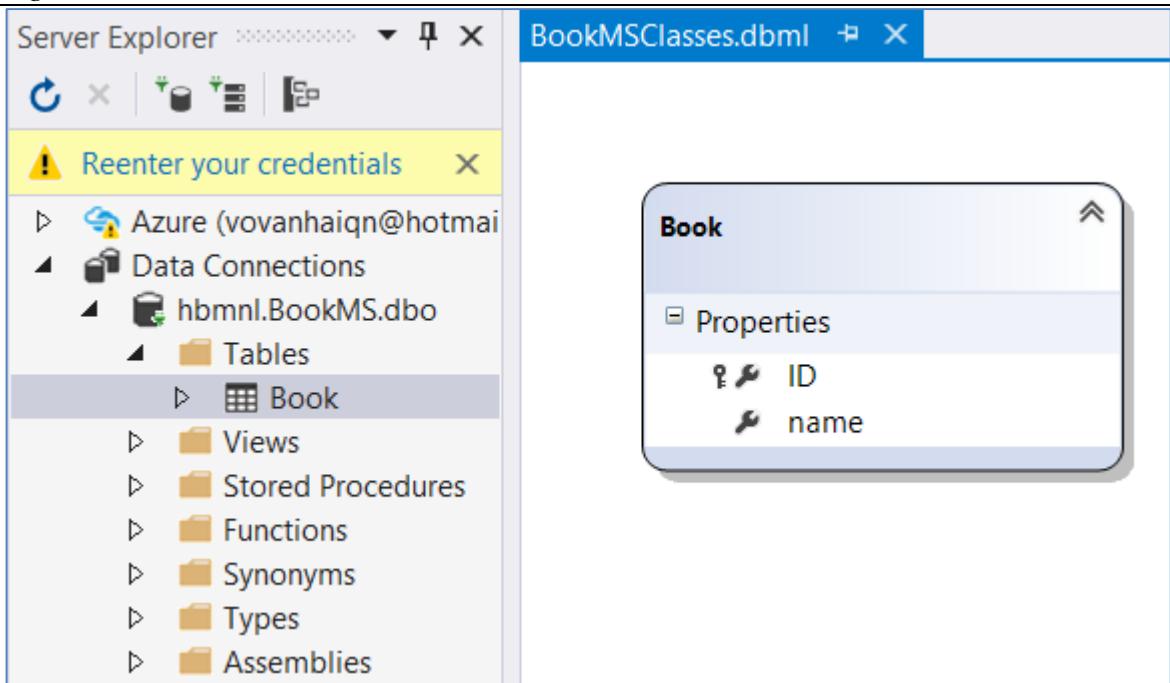


Ở đây ta dùng LINQ để truy xuất cơ sở dữ liệu BookMS đã cho ở trên

Nhảm phải chuột lên Class Library DataAccess, thêm New Item. Chọn LINQ to SQL Classes, Đặt tên BookMSClasses.dbml



Dùng Server Explorer thêm một kết nối đến BookMS rồi kéo vào như hình



Thêm lớp BookDAO có code sau

```
using System.Collections.Generic;
using System.Linq;

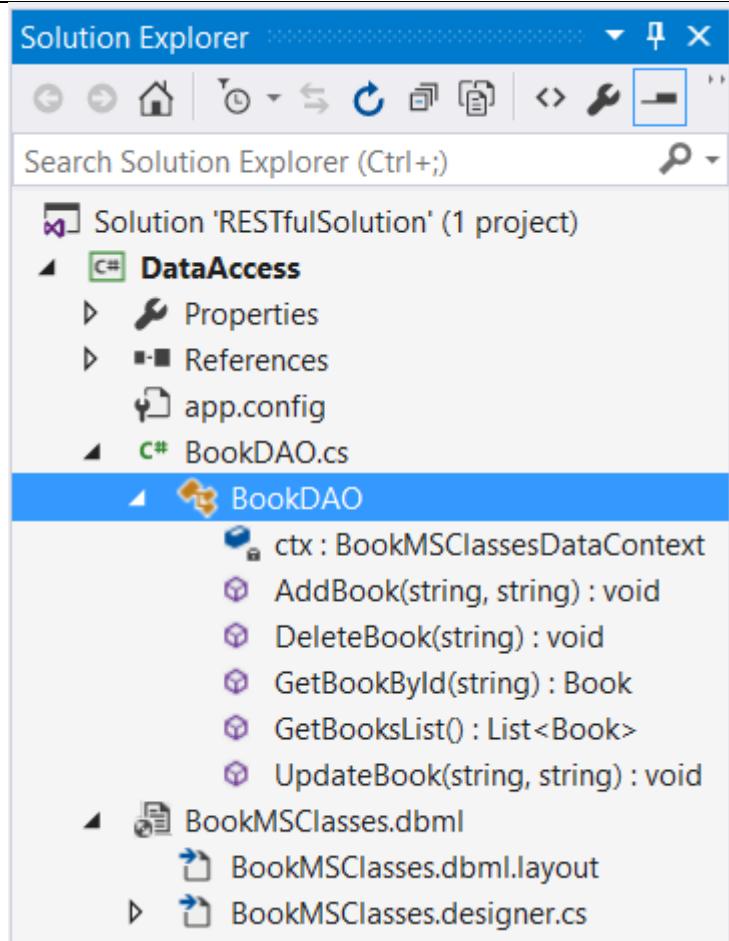
namespace DataAccess
{
    public class BookDAO
    {
        private BookMSClassesDataContext ctx = new BookMSClassesDataContext();
        public void AddBook(string id, string name)
        {
            Book book = new Book { ID = id, name = name };
            ctx.Books.InsertOnSubmit(book);
        }
        public void DeleteBook(string id)
        {
            Book book = new DataAccess.Book { ID = id, name = "" };
            ctx.Books.DeleteOnSubmit(book);
        }

        public Book GetBookById(string id)
        {
            return ctx.Books.SingleOrDefault(book => book.ID == id);
        }

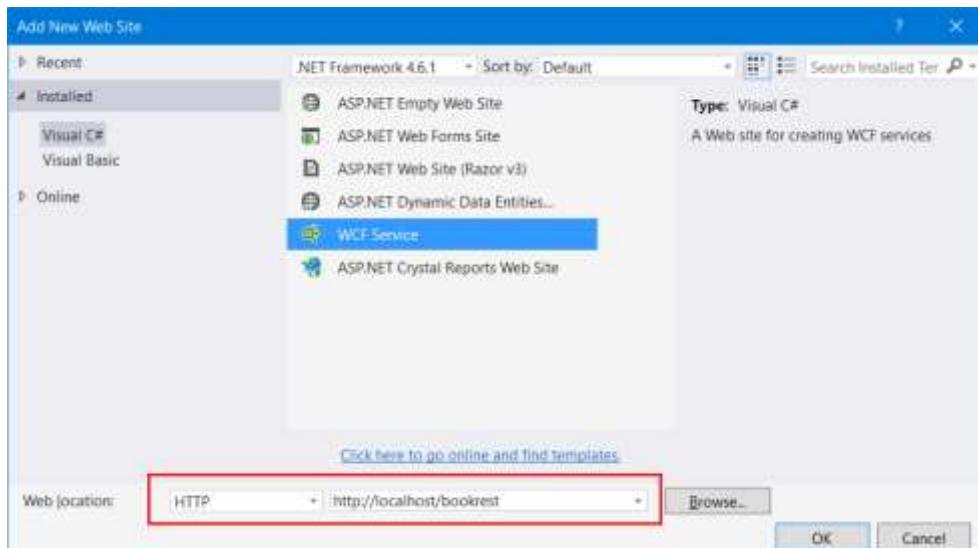
        public List<Book> GetBooksList()
        {
            return ctx.Books.ToList();
        }

        public void UpdateBook(string id, string name)
        {
            Book book = ctx.Books.SingleOrDefault(b => b.ID == id);
            book.name = name;
            ctx.SubmitChanges();
        }
    }
}
```

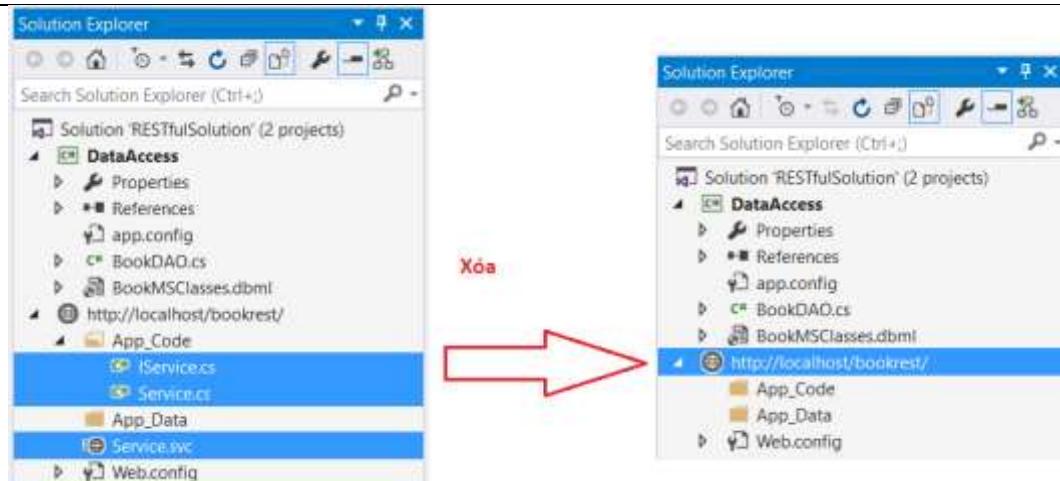
Solution sau bước này có dạng sau



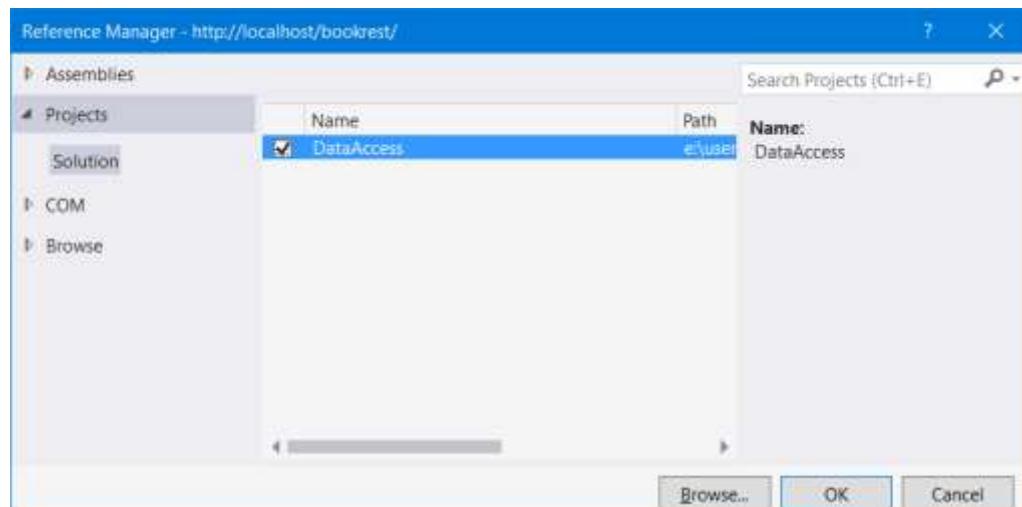
Tiếp theo, ta thêm một WCF WebService



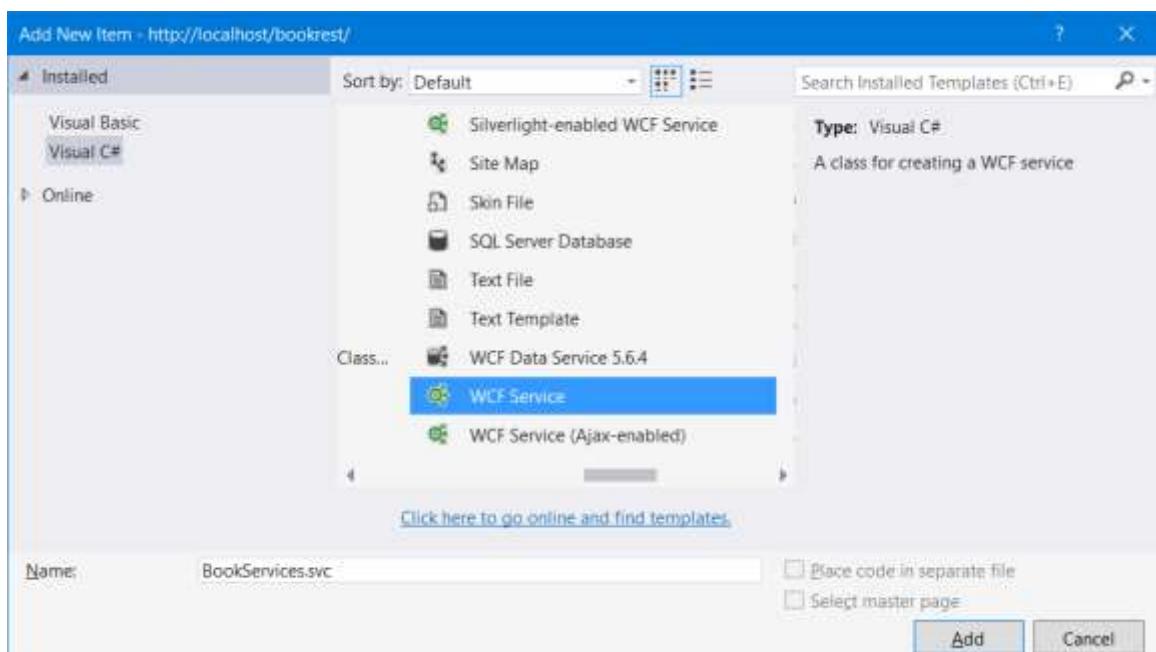
Tiến hành xóa Service svc mà VS tạo sẵn



Thêm tham chiếu để project DataAccess



Nhấn phải chuột lên project, thêm một WCF Service có tên BookServices như hình



Code cho IBookServices

```
using DataAccess;
using System.Collections.Generic;
using System.ServiceModel;
```

```

using System.ServiceModel.Web;

[ServiceContract]
public interface IBookServices
{
    [OperationContract]
    [WebGet]
    List<Book> GetBooksList();

    [OperationContract]
    [WebGet(UriTemplate = "Book/{id}")]
    Book GetBookById(string id);

    [OperationContract]
    [WebInvoke(UriTemplate = "AddBook/{id}/{name}")]
    void AddBook(string id, string name);

    [OperationContract]
    [WebInvoke(UriTemplate = "UpdateBook/{id}/{name}")]
    void UpdateBook(string id, string name);

    [OperationContract]
    [WebInvoke(UriTemplate = "DeleteBook/{id}")]
    void DeleteBook(string id);
}

```

Code cho BookServices implementations

```

using System.Collections.Generic;
using DataAccess;

public class BookServices : IBookServices
{
    private BookDAO dao = new BookDAO();

    public void AddBook(string id, string name)
    {
        dao.AddBook(id, name);
    }

    public void DeleteBook(string id)
    {
        dao.DeleteBook(id);
    }

    public Book GetBookById(string id)
    {
        return dao.GetBookById(id);
    }

    public List<Book> GetBooksList()
    {
        return dao.GetBooksList();
    }

    public void UpdateBook(string id, string name)
    {
        dao.UpdateBook(id, name);
    }
}

```

Web.config

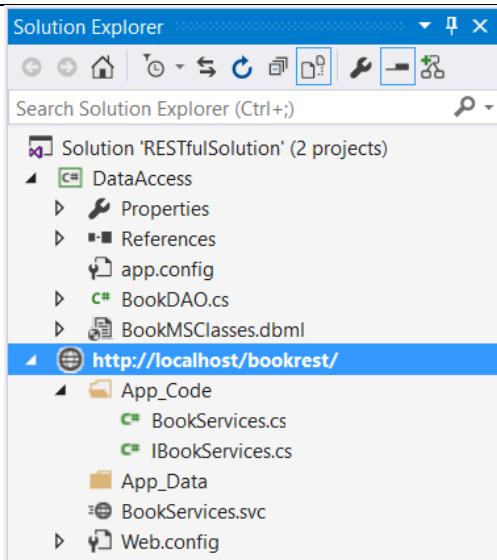
```

<?xml version="1.0"?>
<configuration>
    <appSettings>
        <add key="aspnet:UseTaskFriendlySynchronizationContext" value="true" />

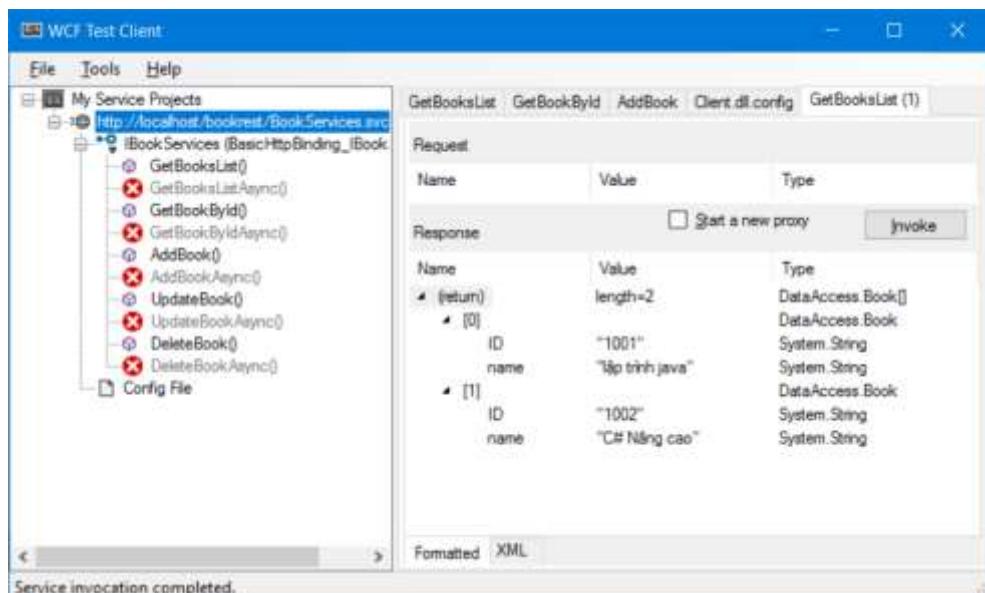
```

```
</appSettings>
<system.web>
    <compilation debug="false" targetFramework="4.6.1" />
    <httpRuntime targetFramework="4.6.1"/>
</system.web>
<system.serviceModel>
    <services>
        <service name="BookServices">
            <endpoint address="" behaviorConfiguration="restfulBehavior"
binding="webHttpBinding" bindingConfiguration="" contract="IBookServices"/>
            <host>
                <baseAddresses>
                    <add baseAddress="http://localhost/bookrest"/>
                </baseAddresses>
            </host>
        </service>
    </services>
    <behaviors>
        <endpointBehaviors>
            <behavior name="restfulBehavior">
                <webHttp/>
            </behavior>
        </endpointBehaviors>
        <serviceBehaviors>
            <behavior>
                <!-- To avoid disclosing metadata information, set the values below to false
before deployment -->
                <serviceMetadata httpGetEnabled="true" httpsGetEnabled="true"/>
                <!-- To receive exception details in faults for debugging purposes, set the
value below to true. Set to false before deployment to avoid disclosing exception
information -->
                <serviceDebug includeExceptionDetailInFaults="false"/>
            </behavior>
        </serviceBehaviors>
    </behaviors>
    <protocolMapping>
        <add binding="basicHttpsBinding" scheme="https" />
    </protocolMapping>
    <serviceHostingEnvironment aspNetCompatibilityEnabled="true"
multipleSiteBindingsEnabled="true" />
</system.serviceModel>
<system.webServer>
    <modules runAllManagedModulesForAllRequests="true"/>
    <directoryBrowse enabled="true"/>
</system.webServer>
</configuration>
```

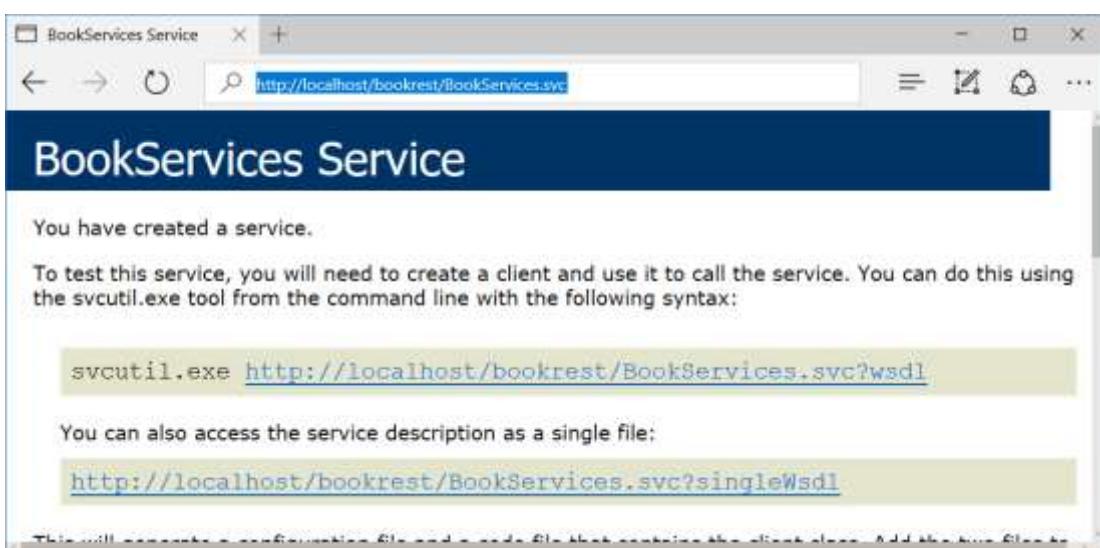
Code sau khi hoàn tất



Thực thi. Test bằng WCF Test Client



Mở bằng Microsoft Edge

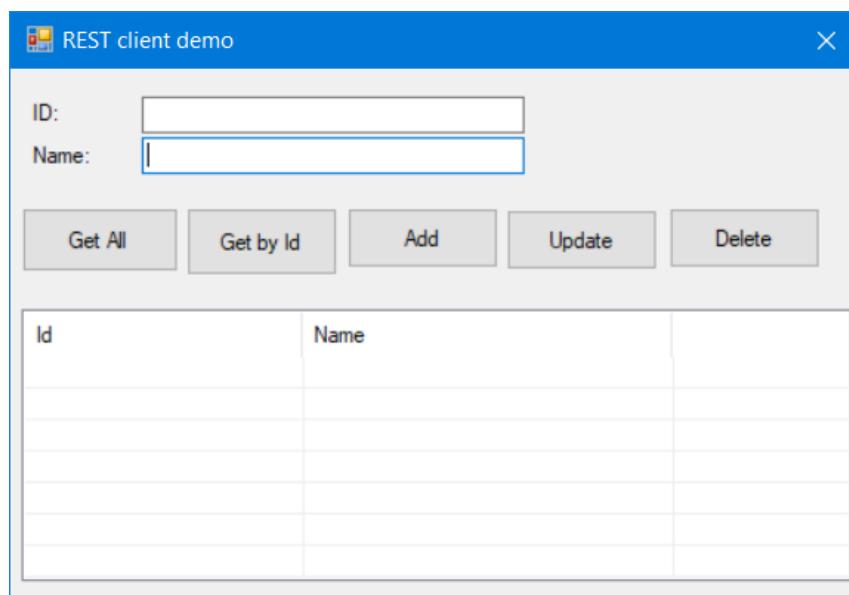


CLIENT

Tạo một Windows Form Application có tên RESTclient.

Thêm tham chiếu đến DataAccess (mục đích lấy đối tượng Book)

Thiết kế giao diện như hình



Khai báo biến cho BASE address của REST

```
private readonly string BASE_URI = "http://localhost/bookrest/BookServices.svc";
```

Code cho nút “Get All”

```
private void getAllButton_Click(object sender, EventArgs e)
{
    string address = BASE_URI + "/GetBooksList";
    WebRequest req = WebRequest.Create(address);
    WebResponse res = req.GetResponse();
    DataContractSerializer dcs = new DataContractSerializer(typeof(Book[]));
    Book[] books = dcs.ReadObject(res.GetResponseStream()) as Book[];
    booksListView.Items.Clear();
    foreach (var book in books)
    {
        ListViewItem li = new ListViewItem(book.ID);
        li.SubItems.Add(book.name);
        booksListView.Items.Add(li);
    }
}
```

Code cho nút “Get by Id”

```
private void getByIdButton_Click(object sender, EventArgs e)
{
    if (IdTextBox.Text.Trim().Equals(""))
        MessageBox.Show("pls enter book id!");
    else
    {
        string address = BASE_URI + "/book/" + IdTextBox.Text;
        WebClient client = new WebClient();
        byte[] data = client.DownloadData(address);
        MemoryStream ms = new MemoryStream(data);
        DataContractSerializer dcs = new DataContractSerializer(typeof(Book));
        Book b = dcs.ReadObject(ms) as Book;
        IdTextBox.Text = b.ID + "";
        nameTextBox.Text = b.name;
    }
}
```

Code cho nút “Add”

Code cho nút “Update”

Code cho nút “Delete”

II. JAX-RS

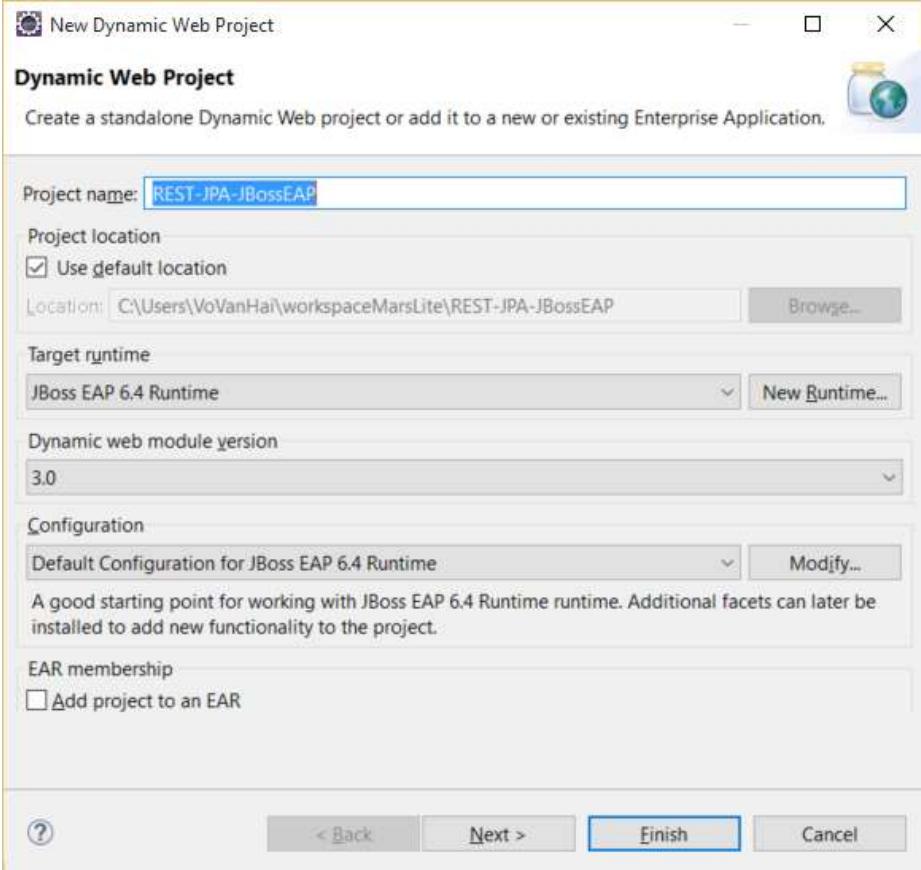
Các Annotation có trong JAX-RS

Sr.No.	Annotation & Description
1	@Path Relative path of the resource class/method.
2	@GET HTTP Get request, used to fetch resource.
3	@PUT HTTP PUT request, used to create resource.
4	@POST HTTP POST request, used to create/update resource.
5	@DELETE HTTP DELETE request, used to delete resource.
6	@HEAD HTTP HEAD request, used to get status of method availability.
7	@Produces States the HTTP Response generated by web service. For example, APPLICATION/XML, TEXT/HTML, APPLICATION/JSON etc.
8	@Consumes States the HTTP Request type. For example, application/x-www-form-urlencoded to accept form data in HTTP body during POST request.
9	@PathParam Binds the parameter passed to the method to a value in path.
10	@QueryParam Binds the parameter passed to method to a query parameter in the path.
11	@MatrixParam Binds the parameter passed to the method to a HTTP matrix parameter in path.
12	@HeaderParam Binds the parameter passed to the method to a HTTP header.
13	@CookieParam Binds the parameter passed to the method to a Cookie.
14	@FormParam Binds the parameter passed to the method to a form value.

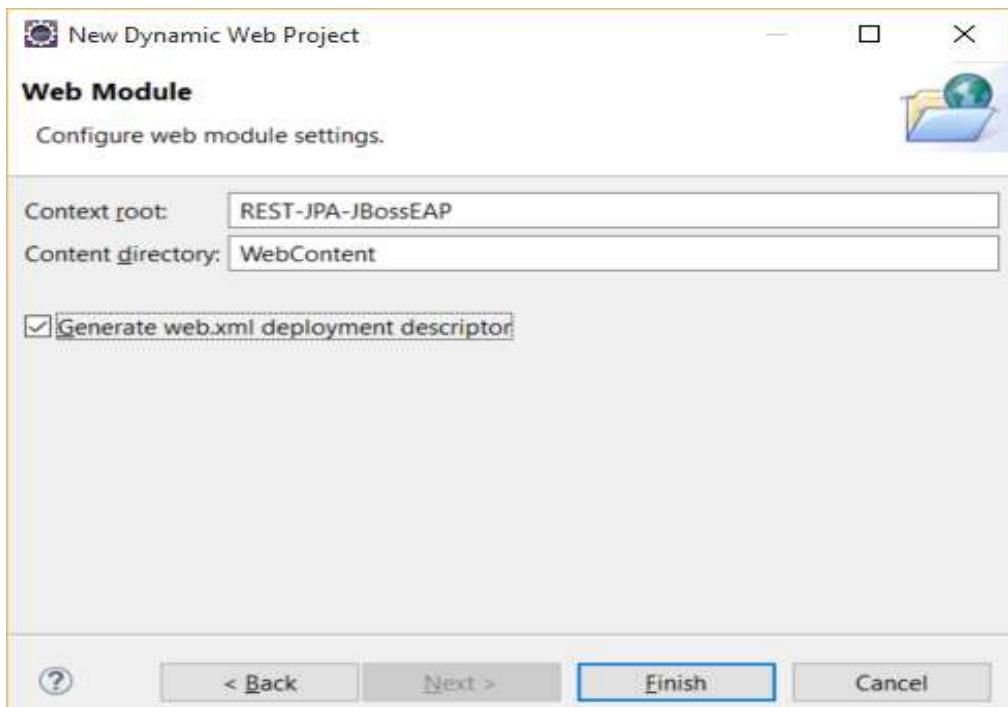
15	@DefaultValue Assigns a default value to a parameter passed to the method.
16	@Context Context of the resource. For example, XMLHttpRequest as a context.

REST - JPA - JBossEAP

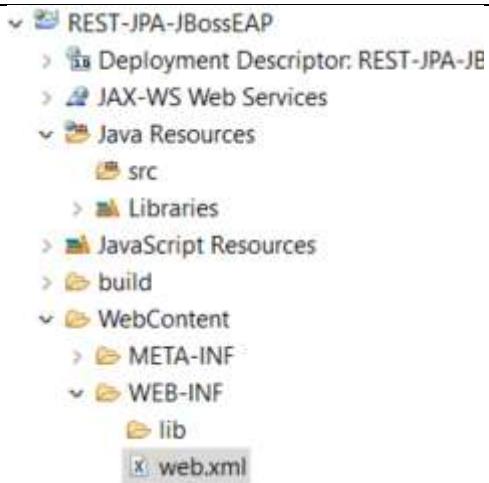
Trong ví dụ này. Chúng ta tạo 1 RESTful web service truy xuất dữ liệu thông qua JPA

TẠO REST

The screenshot shows the 'Dynamic Web Project' configuration window. It includes fields for 'Project name' (REST-JPA-JBossEAP), 'Project location' (using default location at C:\Users\VoVanHai\workspaceMarsLite\REST-JPA-JBossEAP), 'Target runtime' (JBoss EAP 6.4 Runtime), 'Dynamic web module version' (3.0), 'Configuration' (Default Configuration for JBoss EAP 6.4 Runtime), and 'EAR membership' (unchecked). At the bottom are buttons for '?', '< Back', 'Next >', 'Finish' (highlighted in blue), and 'Cancel'.



The screenshot shows the 'Web Module' configuration window. It includes fields for 'Context root' (REST-JPA-JBossEAP) and 'Content directory' (WebContent). A checked checkbox labeled 'Generate web.xml deployment descriptor' is present. At the bottom are buttons for '?', '< Back', 'Next >', 'Finish' (highlighted in blue), and 'Cancel'.



Tổ chức code phân tầng

REST → JPA → DB

Trong đó JPA gồm: entities và các lớp façade.

Tạo entity đặc tả cho một cuốn sách

```
package rest.jpa.entities;

import javax.persistence.Entity;
import javax.persistence.Id;
import javax.persistence.NamedQueries;
import javax.persistence.NamedQuery;
import javax.xml.bind.annotation.XmlRootElement;
import javax.xml.bind.annotation.XmlType;

@Entity//OR mapping
@XmlRootElement//using for transform data
@XmlType(propOrder={"isbn","title","writer","publisher","publishYear"})
@NamedQueries({
    @NamedQuery(name="Book.findAll",query="select b from Book"),
    @NamedQuery(name="Book.findByISBN",query="select b from Book where b.isbn=:isbn"),
    @NamedQuery(name="Book.findByPublisher",query="select b from Book where
b.publisher=:publisher"),
    @NamedQuery(name="Book.findByPublishYear",query="select b from Book where
b.publishYear=?publishYear")
})
public class Book {
    @Id
    private long isbn;
    private String title;
    private String writer;
    private String publisher;
    private int publishYear;
    public Book() {
    }
    public Book(long isbn, String title, String writer, String publisher, int
publishYear) {
        this.isbn = isbn;
        this.title = title;
        this.writer = writer;
        this.publisher = publisher;
        this.publishYear = publishYear;
    }
    public long getIsbn() {
        return isbn;
    }
}
```

```

    }
    public void setIsbn(long isbn) {
        this.isbn = isbn;
    }
    public String getTitle() {
        return title;
    }
    public void setTitle(String title) {
        this.title = title;
    }
    public String getWriter() {
        return writer;
    }
    public void setWriter(String writer) {
        this.writer = writer;
    }
    public String getPublisher() {
        return publisher;
    }
    public void setPublisher(String publisher) {
        this.publisher = publisher;
    }
    public int getPublishYear() {
        return publishYear;
    }
    public void setPublishYear(int publishYear) {
        this.publishYear = publishYear;
    }
    @Override
    public int hashCode() {
        final int prime = 31;
        int result = 1;
        result = prime * result + (int) (isbn ^ (isbn >>> 32));
        return result;
    }
    @Override
    public boolean equals(Object obj) {
        if (this == obj)
            return true;
        if (obj == null)
            return false;
        if (getClass() != obj.getClass())
            return false;
        Book other = (Book) obj;
        if (isbn != other.isbn)
            return false;
        return true;
    }
    @Override
    public String toString() {
        return "Book [isbn=" + isbn + ", title=" + title + ", writer=" + writer + ",
publisher=" + publisher
                + ", publishYear=" + publishYear + "]";
    }
}

```

Tạo file persistence.xml trong thư mục META-INF

```

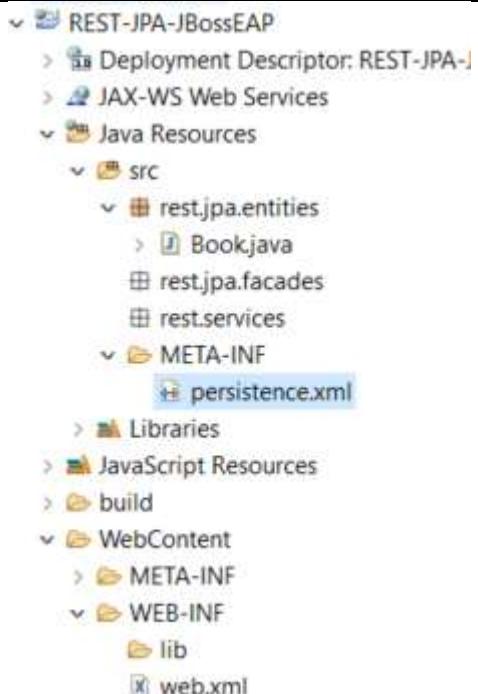
<?xml version="1.0" encoding="UTF-8"?>
<persistence version="2.1" xmlns="http://xmlns.jcp.org/xml/ns/persistence"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/persistence
        http://xmlns.jcp.org/xml/ns/persistence/persistence_2_1.xsd">

```

```

<persistence-unit name="TeoPU" transaction-type="JTA">
    <provider>org.hibernate.ejb.HibernatePersistence</provider>
    <jta-data-source>BookDS</jta-data-source>
    <properties>
        <property name="hibernate.dialect"
                  value="org.hibernate.dialect.SQLServerDialect"/>
        <property name="hibernate.hbm2ddl.auto" value="update"/>
        <property name="hibernate.show_sql" value="true"/>
    </properties>
</persistence-unit>
</persistence>

```



Cấu hình web.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
          xmlns="http://java.sun.com/xml/ns/javaee"
          xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
          http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd"
          id="WebApp_ID" version="3.0">
    <display-name>REST-JPA-JBossEAP</display-name>
    <welcome-file-list>
        <welcome-file>index.html</welcome-file>
        <welcome-file>index.jsp</welcome-file>
    </welcome-file-list>

    <servlet-mapping>
        <servlet-name>javax.ws.rs.core.Application</servlet-name>
        <url-pattern>/rest/*</url-pattern>
    </servlet-mapping>

    <persistence-context-ref>
        <persistence-context-ref-name>persistence/em</persistence-context-ref-name>
        <persistence-unit-name>BookPU</persistence-unit-name>
    </persistence-context-ref>
</web-app>

```

Tạo lớp façade để control JPA entities

```
package rest.jpa.facades;
```

```

import java.util.List;
import javax.ejb.LocalBean;
import javax.ejb.Stateless;
import javax.persistence.EntityManager;
import javax.persistence.PersistenceContext;
import javax.persistence.PersistenceContextType;
import rest.jpa.entities.Book;

@Stateless
@LocalBean
public class BookFacade {
    @PersistenceContext(unitName="BookPU", type=PersistenceContextType.TRANSACTION)
    private EntityManager em;

    public void insert(Book b){
        em.persist(b);
    }
    public void delete(Book b){
        em.remove(em.merge(b));
    }
    public void update(Book b){
        em.merge(b);
    }

    public List<Book> getAll(){
        return em.createNamedQuery("Book.findAll").getResultList();
    }

    public Book getByISBN(long isbn){
        return (Book) em.createNamedQuery("Book.findByISBN")
            .setParameter("isbn", isbn)
            .getSingleResult();
    }

    public List<Book> getByPublisher(String publisher){
        return em.createNamedQuery("ok.findByPublisher")
            .setParameter("publisher", publisher)
            .getResultList();
    }
    public List<Book> getByPublishYear(int year){
        return em.createNamedQuery("Book.findByPublishYear")
            .setParameter("publishYear", year)
            .getResultList();
    }
}

```

Tạo Rest service

```

package rest.services;

import java.util.List;
import javax.ejb.EJB;
import javax.ejb.LocalBean;
import javax.ejb.Stateless;
import javax.ws.rs.Consumes;
import javax.ws.rs.DELETE;
import javax.ws.rs.GET;
import javax.ws.rs.POST;
import javax.ws.rs.PUT;
import javax.ws.rs.Path;
import javax.ws.rs.PathParam;
import javax.ws.rs.Produces;

```

```
import javax.ws.rs.core.MediaType;
import rest.jpa.entities.Book;
import rest.jpa.facades.BookFacade;

@Stateless
@LocalBean
@Path("/books")
public class BookService {
    @EJB
    private BookFacade bf;
    public BookService() {
    }

    @GET
    @Produces({MediaType.APPLICATION_JSON})
    public List<Book> getAll(){
        return bf.getAll();
    }

    @GET
    @Path("{isbn}")
    @Produces({MediaType.APPLICATION_XML})
    public Book getByISBN(@PathParam("isbn") long isbn){
        return bf.getByISBN(isbn);
    }

    @GET
    @Path("/findByPublisher/{publisher}")
    @Produces({MediaType.APPLICATION_XML})
    public List<Book> getByPublisher(
            @PathParam("publisher") String publisher){
        return bf.getByPublisher(publisher);
    }

    @GET
    @Path("/getByPublishYear/{year}")
    @Produces({MediaType.APPLICATION_XML})
    public List<Book> getByPublishYear(
            @PathParam("year") int year){
        return bf.getByPublishYear(year);
    }

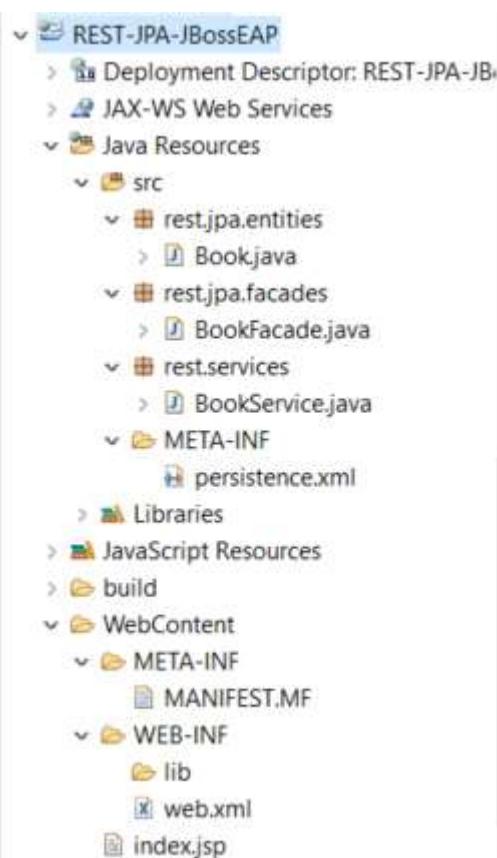
    @PUT
    // @Path("/insert")
    @Consumes({MediaType.APPLICATION_XML,MediaType.APPLICATION_JSON})
    public void insert(Book b){
        bf.insert(b);
    }

    @DELETE
    // @Path("/delete")
    @Consumes({MediaType.APPLICATION_XML,MediaType.APPLICATION_JSON})
    public void delete(Book b){
        bf.delete(b);
    }

    @POST
    // @Path("update")
    @Consumes({MediaType.APPLICATION_XML,MediaType.APPLICATION_JSON})
    public void update(Book b){
        bf.update(b);
    }
}
```

}

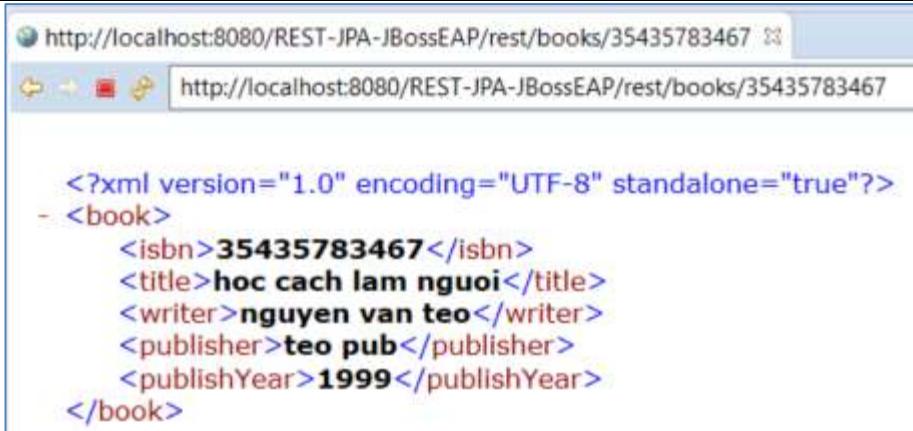
Cấu trúc chương trình như sau:



BookFacade.java BookService... web.xml persistence.xml CustomerSer... http://local...

<http://localhost:8080/REST-JPA-JBossEAP/rest/books/>

```
<?xml version="1.0" encoding="UTF-8" standalone="true"?>
- <collection>
  - <book>
    <isbn>3567</isbn>
    <title>dxkjgjfdgjk</title>
    <writer>ty</writer>
    <publisher>zdkjgsdg</publisher>
    <publishYear>2000</publishYear>
  </book>
  - <book>
    <isbn>35435783467</isbn>
    <title>hoc cach lam nguoi</title>
    <writer>nguyen van teo</writer>
    <publisher>teo pub</publisher>
    <publishYear>1999</publishYear>
  </book>
</collection>
```



CLIENT

Client code sử dụng HttpURLConnection

```

package console.rest.client;

import java.io.OutputStream;
import java.io.StringWriter;
import java.net.HttpURLConnection;
import java.net.URL;
import java.util.Scanner;
import javax.ws.rs.core.MediaType;
import javax.xml.bind.JAXBContext;
import javax.xml.bind.Marshaller;
import rest.jpa.entities.Book;

public class ConsoleRESTclient {
    public static void main(String[] args) throws Exception{
        /*Book b=new Book(3567, "Tu hoc JAvA", "truong van coi", "teo pub", 1999);
        insert(b);
        update(b);*/
        //delete(b);
        get();
    }

    static void get()throws Exception{
        URL url=new URL("http://localhost:8080/REST-JPA-JBossEAP/rest/books/");

        HttpURLConnection con=(HttpURLConnection)url.openConnection();
        //qui ding cac tinh chat cho connection
        con.setDoOutput(true); //cho phep gui
        con.setDoInput(true); //cho phep nhan
        con.setRequestProperty("Content-Type", MediaType.APPLICATION_XML);

        con.setRequestMethod("GET");

        //doc ket qua tu server
        Scanner in=new Scanner(con.getInputStream());
        while(in.hasNextLine()){
            String line=in.nextLine();
            System.out.println(line);
        }
        in.close();
    }
    static void get(long isbn) throws Exception{
        URL url=new URL("http://localhost:8080/REST-JPA-JBossEAP/rest/books/"+isbn);

        HttpURLConnection con=(HttpURLConnection)url.openConnection();
    }
}

```

```

//qui ding cac tinh chat cho connection
con.setDoOutput(true); //cho phep gui
con.setDoInput(true); //cho phep nhan
con.setRequestProperty("Content-Type", MediaType.TEXT_PLAIN);

con.setRequestMethod("GET");

//doc ket qua tu server
Scanner in=new Scanner(con.getInputStream());
while(in.hasNextLine()){
    String line=in.nextLine();
    System.out.println(line);
}
in.close();

}

static void insert(Book b) throws Exception{
URL url=new URL("http://localhost:8080/REST-JPA-JBossEAP/rest/books/");

HttpURLConnection con=(HttpURLConnection)url.openConnection();
//qui ding cac tinh chat cho connection
con.setDoOutput(true); //cho phep gui
con.setDoInput(true); //cho phep nhan
con.setRequestProperty("Content-Type", "application/xml");
con.setRequestMethod("PUT");

JAXBContext ctx=JAXBContext.newInstance(Book.class);
Marshaller ms = ctx.createMarshaller();
StringWriter writer=new StringWriter();
ms.marshal(b, writer);
String xml=writer.toString();

//gui object can luu len
OutputStream os=con.getOutputStream();
os.write(xml.getBytes());
os.flush();

//lay response code de chac chan la no da submit
int resCode = con.getResponseCode();
System.out.println(resCode); //nam trong [200,300] la ok, thuong la 204
}

static void update(Book b) throws Exception{
URL url=new URL("http://localhost:8080/REST-JPA-JBossEAP/rest/books/");

HttpURLConnection con=(HttpURLConnection)url.openConnection();
//qui ding cac tinh chat cho connection
con.setDoOutput(true); //cho phep gui
con.setDoInput(true); //cho phep nhan
con.setRequestProperty("Content-Type", "application/xml");
con.setRequestMethod("POST");

JAXBContext ctx=JAXBContext.newInstance(Book.class);
Marshaller ms = ctx.createMarshaller();
StringWriter writer=new StringWriter();
ms.marshal(b, writer);
String xml=writer.toString();

//gui object can luu len
OutputStream os=con.getOutputStream();

```

```
os.write(xml.getBytes());
os.flush();

//lay response code de chac chan la no da submit
int resCode = con.getResponseCode();
System.out.println(resCode); //nam trong [200,300] la ok, thuong la 204
}

static void delete(Book b) throws Exception{
URL url=new URL("http://localhost:8080/REST-JPA-JBossEAP/rest/books/");

HttpURLConnection con=(HttpURLConnection)url.openConnection();
//qui ding cac tinh chat cho connection
con.setDoOutput(true); //cho phep gui
con.setDoInput(true); //cho phep nhan
con.setRequestProperty("Content-Type", "application/xml");
con.setRequestMethod("DELETE");

JAXBContext ctx=JAXBContext.newInstance(Book.class);
Marshaller ms = ctx.createMarshaller();
StringWriter writer=new StringWriter();
ms.marshal(b, writer);
String xml=writer.toString();

//gui object can luu len
OutputStream os=con.getOutputStream();
os.write(xml.getBytes());
os.flush();

//lay response code de chac chan la no da submit
int resCode = con.getResponseCode();
System.out.println(resCode); //nam trong [200,300] la ok, thuong la 204
}
}
```

III. REST VỚI NODEJS

IV. SECURITY