

# TẠO WEB SERVICES & GỌI WEB SERVICES TỪ DI ĐỘNG ANDROID



**Cao Tuấn Dũng**

# NỘI DUNG



- I. Tạo web services sử dụng myEclipse
- II. Deploy webservises
- III. Gọi web services từ di động android

# I. Tạo webservices sử dụng myEclipse



- myEclipse hỗ trợ tạo web services với nhiều frame work khác nhau (XFIRE, JAX-WS ...v.v.) tutorial này chỉ đề cập đến việc xây dựng web services sử dụng frame work JAX-WS.
- Phần này sẽ bao gồm:
  - 1.1 Khởi tạo project
  - 1.2 Viết các method, class thực thi
  - 1.3 Build web services (Gen web services bằng myEclipse)

# 1.1 Khởi tạo project



- Create web services project

**New Web Service Project**

Web service project creation details

Web Project Details

Project Name:

Location: ☒ Use default location

Directory:

Source folder:

Web root folder:

Context root URL:

Maven

☐ Add Maven support

[Learn more about Maven4MyEclipse...](#)

Web Service & J2EE Details

Framework: ☒ JAX-WS ☐ REST (JAX-RS) ☐ XFire ([deprecated](#))

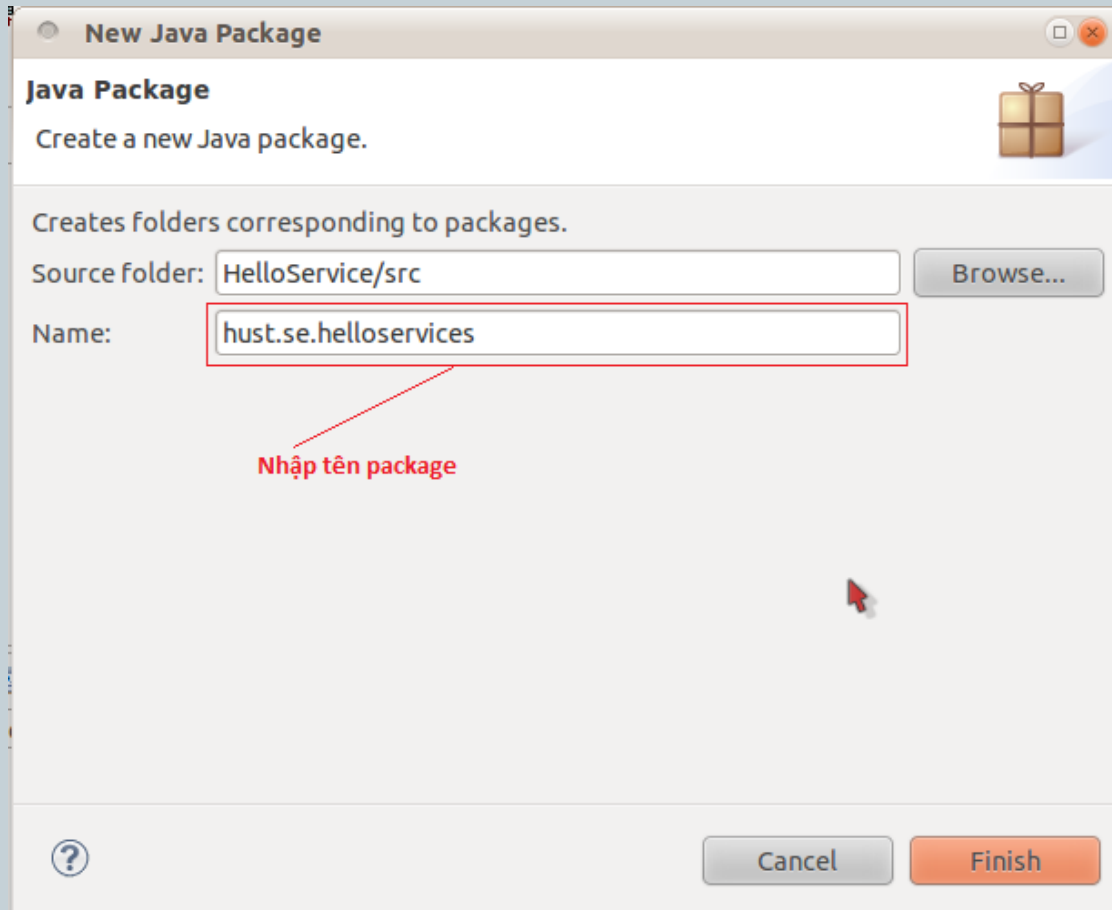
J2EE specification: ☒ Java EE 5.0 ☐ J2EE 1.4 ☐ J2EE 1.3

*Chon JAX-WS frame work*

# 1.1 Khởi tạo project



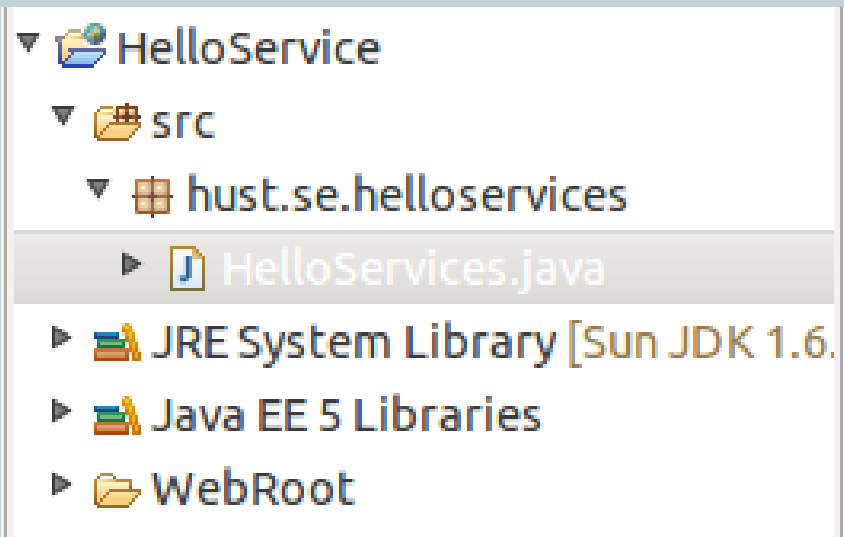
- Tạo package



## 1.1 Khởi tạo project



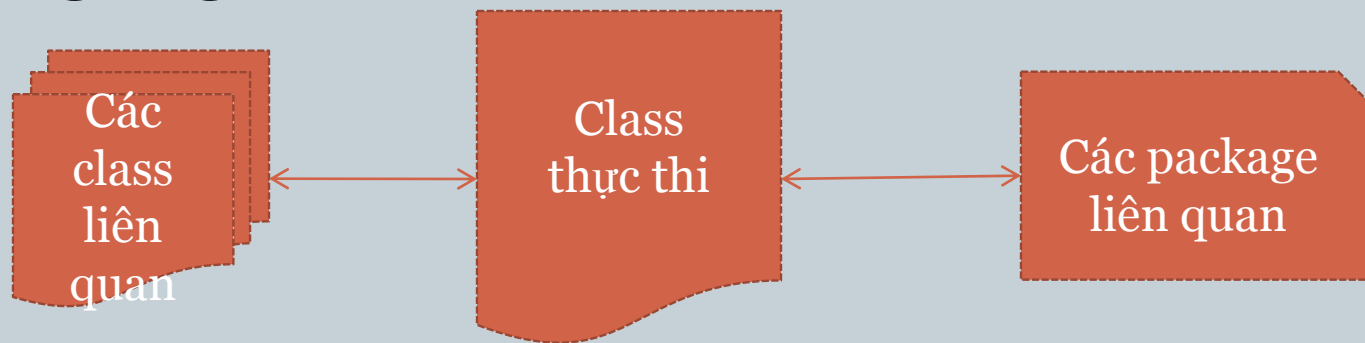
- Tạo lớp HelloService (Thực thi các method của service). Sau khi hoàn tất cấu trúc thư mục project sẽ như sau:



## 1.2 Viết các method thực thi



- Viết tất cả các hàm thực thi trong cùng một lớp (Lớp này có thể gọi đến nhiều lớp, gói khác trong project để thực hiện nhiệm vụ của mình mà không cần build các lớp đó thành services). Trong ví dụ này chúng ta chỉ viết một method đơn giản sayHello(String yourName). Hàm này sẽ trả về một string là câu chào tương ứng.



## 1.2 Viết các method thực thi



- Class HelloServices với hàm đơn giản sayHello

```
HelloServices.java ✕
2 /**
3  * This class used to building webservice
4  * @author mrhien
5  */
6 public class HelloServices {
7     /**
8      * Constructor
9      */
10    public HelloServices() {
11    }
12    /**
13     * Các method (sẽ được chuyển thành service để sử dụng)
14     */
15    public String sayHello(String yourName){
16        return "Welcome "+yourName +" to webservices!";
17    }
18 }
```



## 1.3 Build web services



- Chọn File -> new Web services (Chú ý tick chọn create from Java Class và JAX-WS Framework. Click

**New Web Service**

Select web service framework & creation scenario

Project:

[Create a new web service project...](#)

Info: Java EE 5 Web project

Framework: ☒ JAX-WS  
☐ REST (JAX-RS)  
☐ XFire ([deprecated](#))

Strategy: ☐ Create web service from WSDL document (Top-down scenario)  
☒ Create web service from Java class (Bottom-up scenario)

## 1.3 Build web services



- Browse lớp thực thi HelloServices chọn Generate WSDL

**New Web Service**

**New SOAP Web Service - Bottom-up Scenario**

Create web service from Java class

Web service project: HelloService

Java Implementation

Java class:

Delegate class name:

SOAP & Servlet Configuration

SOAP style/use:

☐ Enable SOAP 1.2 Binding

☐ Enable MTOM Support

☒ Generate WSDL in project

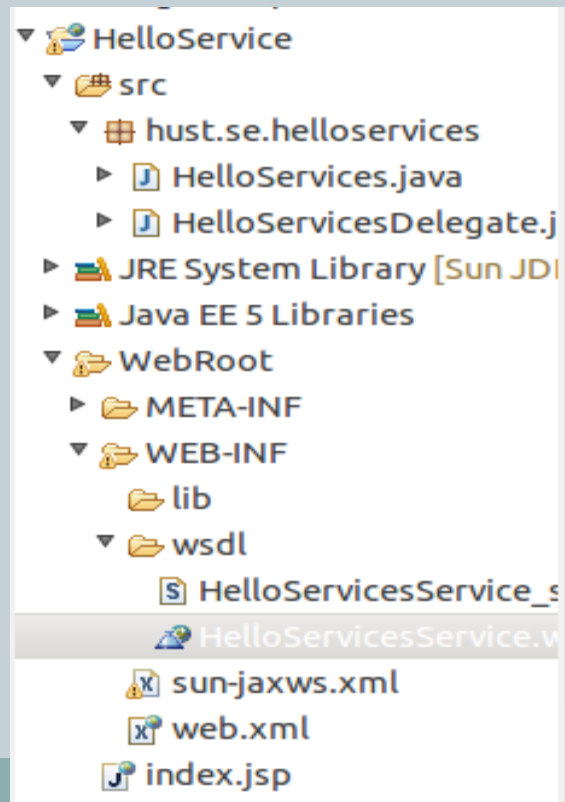
WSDL Customization

Target namespace:

## 1.3 Build Web services



- Sau khi hoàn tất myEclipse tự động sinh ra:
  - Lớp HelloServicesDelegate (Khai báo web services):
  - File HelloServices.WSDL (file đặc tả web services)



## 1.3 Build web services



- HelloServicesDelegate.java

```
1 package hust.se.helloservices;
2
3 @javax.jws.WebService([
4     targetNamespace = "http://helloservices.se.hust/",
5     serviceName = "HelloServicesService",
6     portName = "HelloServicesPort",
7     wsdlLocation = "WEB-INF/wsdl/HelloServicesService.wsdl")
8 public class HelloServicesDelegate {
9
10     hust.se.helloservices.HelloServices helloServices = new hust.se.helloservices.HelloServices();
11
12     public String sayHello(String yourName) {
13         return helloServices.sayHello(yourName);
14     }
15
16 }
```

## 1.3 Build web services



- HelloServicesService.wsdl
- Cần chú ý trong file này chính là **Endpoint** được tạo



```
17     <input message="tns:sayHello"/>
18     <output message="tns:sayHelloResponse"/>
19   </operation>
20 </portType>
21 <binding name="HelloServicesPortBinding" type="tns:HelloServicesDelegate">
22   <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
23   <operation name="sayHello">
24     <soap:operation soapAction=""/>
25     <input>
26       <soap:body use="literal"/>
27     </input>
28     <output>
29       <soap:body use="literal"/>
30     </output>
31   </operation>
32 </binding>
33 <service name="HelloServicesService">
34   <port binding="tns:HelloServicesPortBinding" name="HelloServicesPort">
35     <soap:address location="http://localhost:8080/HelloService/HelloServicesPort"/>
36   </port>
37 </service>
38 </definitions>
39
```

## II. Deploy web services



- Sau khi một web services được tạo ra. Để có thể sử dụng được chúng ta cần deploy nó. Có 2 cách chính để deploy:
  - Sử dụng myEclipse cho deploy tự động.
    - ✦ Click phải vào **project**, chọn **run as server application**. Chọn server mà project được deploy trên đó là được.
  - Deploy bằng tay.
    - ✦ Export cả project trên ra file **\*.war** (click phải vào project, chọn export, chọn export ra file war)
    - ✦ Copy file war vào thư mục **/webapps** của Tomcat server hoặc thư mục tương tự đối với các server khác.
    - ✦ Chạy Tomcat server. Server sẽ tự động nhận và deploy services.

## II. Deploy web services



- Test web services:
  - Sử dụng myEclipse browser để test webservice vừa deploy

Web Services Explorer

Navigator

WSDL Main

Actions

**Open WSDL**

Enter the URL of a WSDL document and click **Go** to explore.

WSDL URL [Browse...](#)

**Go** **Reset**

Click đây để mở giao diện gọi web services thông qua WSDL URL

Nhập địa chỉ WSDL vào đây. Chính là endpoint ở phần cuối trong file wsdl + "?WSDL"

## II. Deploy web services



- Test web services:
  - Nhân Go để duyệt web services



### Open WSDL

Enter the URL of a WSDL document and click **Go** to explore.

WSDL URL [Browse...](#)



## II. Deploy web services



- Test web services:
  - Sau khi deploy thành công sẽ có thông báo ở phần status.
  - Method được cung cấp sẽ được hiển thị ở phần Operations

**Operations**

Name	Documentation
<a href="#">sayHello</a>	-

**Endpoints** [Add](#) [Remove](#)

Endpoints
<input type="checkbox"/> <a href="http://localhost:8080/HelloService/HelloServicesPort">http://localhost:8080/HelloService/HelloServicesPort</a>

**i Status**

IWAB0381I http://localhost:8080/HelloService/HelloServicesPort?WSDL was successfully opened.

## II. Deploy web services



- Chọn hàm operator sayHello, nhập vào tham số phù hợp, nhấn Go để gọi web services

Endpoints

▼ **Body**

▼ [sayHello](#)


▼ [arg0](#) string [Add](#) [Remove](#)

<input type="checkbox"/>	Values
<input type="checkbox"/>	Phan Hiền


## II. Deploy web services




- Kết quả trả về sẽ ở phần status


 Status

[Source](#)

 **Body**

 sayHelloResponse

return (string): Welcome Phan Hiến to webservices!



# III. Gọi và sử dụng web services từ android



- Muốn sử dụng được web services ta phải biết được:
  - **EndPoint URL**: Link tới thành phần mô tả web services (cụ thể ở đây là file wsdl). Có thể biết thông qua file wsdl.

```
<service name="HelloServicesService">
  <port binding="tns:HelloServicesPortBinding" name="HelloServicesPort">
    <soap:address location="http://localhost:8080/HelloService/HelloServicesPort"/>
  </port>
</service>
```

- **Namespace**: là base URL được sinh ra (xem trong file wsdl hoặc file delegate đều có)
- **Method name**: tên của method.

```
package hust.se.helloservices;

@javax.jws.WebService(
    targetNamespace = "http://helloservices.se.hust/",
    serviceName = "HelloServicesService",
    portName = "HelloServicesPort",
    wsdlLocation = "WEB-INF/wsdl/HelloServicesService.wsdl")
```

# III. Gọi và sử dụng web services từ android



- Trong tutorial sẽ không hướng dẫn việc làm thế nào để xây dựng ứng dụng trên android mà chỉ chú trọng những phần nào liên quan đến việc gọi web services.
- Tài liệu về việc phát triển 1 android app có thể tham khảo ở các link sau:
  - <http://developer.android.com/index.html>
  - <http://vietandroid.com/ca-c-ba-i-ha-c-ca-ka-m-ma-ngua-n/1192-hae-ng-da-n-la-p-tra-nh-cae-ba-n-va-i-android-ba-i-o-a.html>
- Phần này sẽ bao gồm:
  - Khởi tạo project
  - Sử dụng web services
  - Demo

# III. Gọi và sử dụng web services từ android



- Khởi tạo project: File → new android project

**New Android Project**

**New Android Project**

✖ An SDK Target must be specified.

Project name:

Contents

☒ Create new project in workspace

☐ Create project from existing source

☒ Use default location

Location:

☐ Create project from existing sample

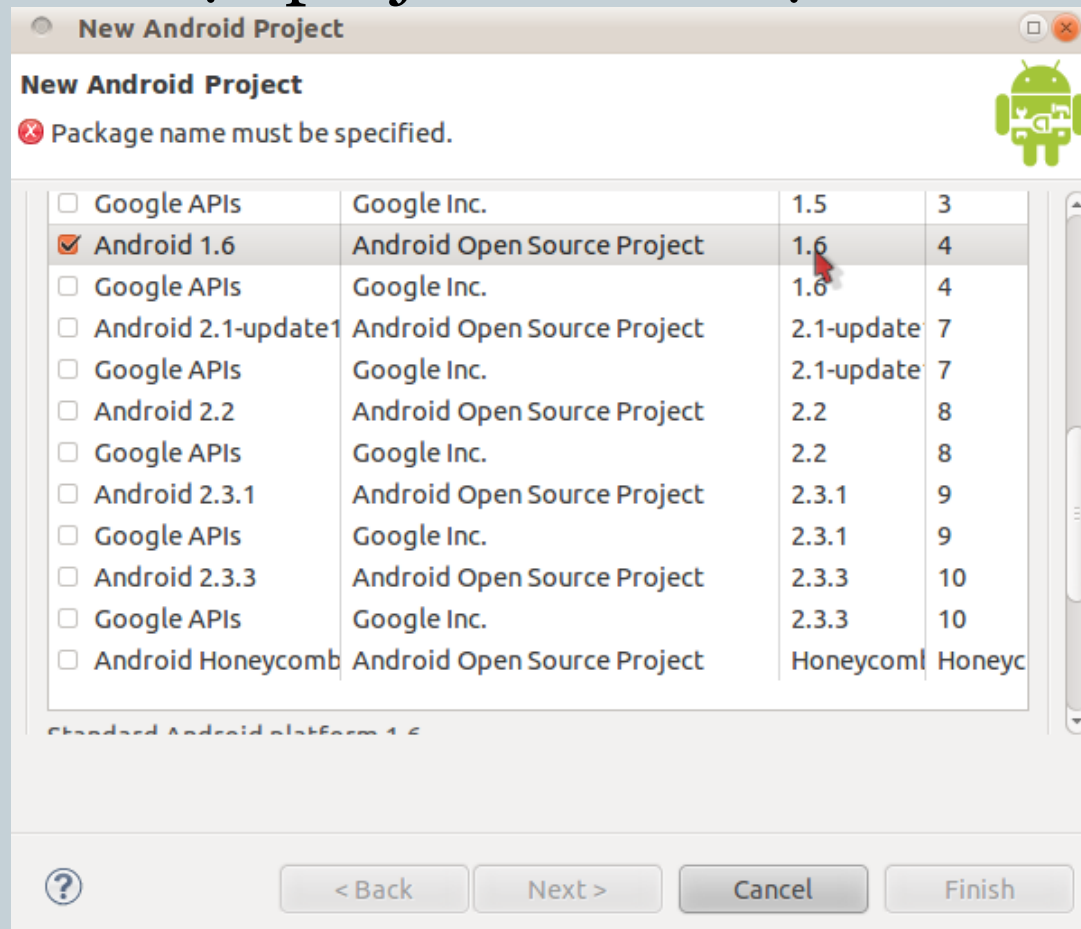
Samples:

Build Target

# III. Gọi và sử dụng web services từ android



- Khởi tạo project: Tick chọn API level



# III. Gọi và sử dụng web services từ android



- Khởi tạo project: Tạo tên gói, Activity chính

**New Android Project**

Creates a new Android Project resource.

<input type="checkbox"/> Google APIs	Google Inc.	2.3.1	9
<input type="checkbox"/> Android 2.3.3	Android Open Source Project	2.3.3	10
<input type="checkbox"/> Google APIs	Google Inc.	2.3.3	10
<input type="checkbox"/> Android Honeycomb	Android Open Source Project	Honeycomb	Honeycomb

Standard Android platform 1.6

**Properties**

Application name:

Package name:

☒ Create Activity:

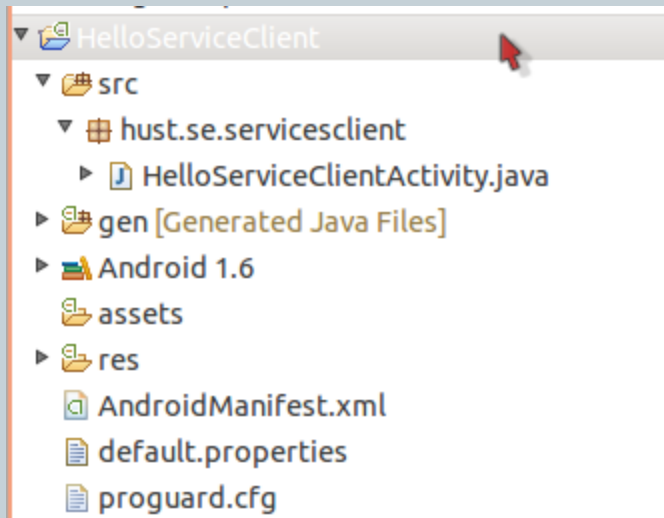
Min SDK Version:



# III. Gọi và sử dụng web services từ android



- Khởi tạo project: sau khi hoàn tất



# III. Gọi và sử dụng web services từ android



- Khởi tạo project:
  - Để ứng dụng của bạn có thể sử dụng được các web services, phải khai báo use-permission trong file manifest

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="hust.se.servicesclient"
    android:versionCode="1"
    android:versionName="1.0">
    <uses-sdk android:minSdkVersion="4" />

    <application android:icon="@drawable/icon" android:label="@string/app_name">
        <activity android:name=".HelloServiceClientActivity"
            android:label="@string/app_name">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

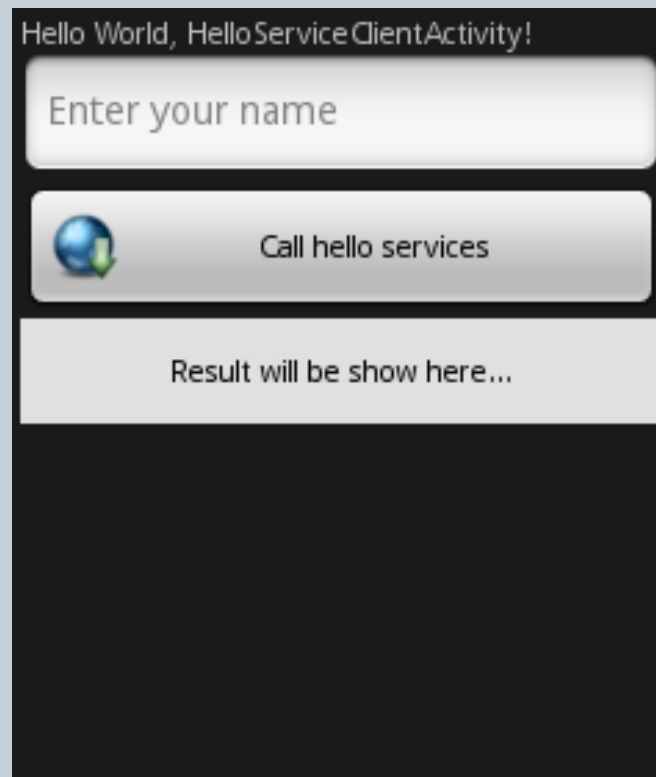
    <uses-permission android:name="android.permission.INTERNET"></uses-permission>

</manifest>
```

# III. Gọi và sử dụng web services từ android



- Khởi tạo project:
  - Tạo giao diện đơn giản như sau (trong file **main.xml**)



# III. Gọi và sử dụng web services từ android



- Sử dụng web services:
  - Lấy **IP** của máy chủ (nếu deploy trên server thật thì có thể sử dụng thay bằng host name):

```
mrhien@mrhien-HP-ProBook-4410s:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 00:25:b3:4c:58:7c
          inet addr:192.168.1.12  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::225:b3ff:fe4c:587c/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:81089 errors:0 dropped:0 overruns:0 frame:0
          TX packets:57886 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:103663613 (103.6 MB)  TX bytes:6160856 (6.1 MB)
          Interrupt:17

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:499886 errors:0 dropped:0 overruns:0 frame:0
          TX packets:499886 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:411392833 (411.3 MB)  TX bytes:411392833 (411.3 MB)
```

# III. Gọi và sử dụng web services



- Sử dụng web services:
  - Khai báo các endpoint, namespace và method name.
  - **Chú ý:** Trong Endpoint URL phải thay “localhost” bằng địa chỉ IP máy chủ (lý do là vì localhost trên android không phải là localhost của máy chủ)

```
public class HelloServiceClientActivity extends Activity {  
  
    private String ENDPOINT_URL      = "http://192.168.1.12:8080/HelloService/HelloServicesPort?WSDL";  
    private String NAMESPACE        = "http://helloservices.se.hust/";  
    private String METHOD_NAME     = "sayHello";  
  
    /** Called when the activity is first created. */  
    @Override  
    public void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.main);  
    }  
}
```

# III. Gọi và sử dụng web services



- Sử dụng web services:
  - Lấy về các thành phần giao diện

```
/**  
 * Lay ve cac thanh phan giao dien  
 */  
EditText nameEditText = (EditText) findViewById(R.id.nameEditText);  
Button callServicesButton = (Button) findViewById(R.id.callServicesButton);  
TextView resultTextView = (TextView) findViewById(R.id.resultTextView);
```

- Mọi xử lý sẽ được thực hiện khi click button “call services”

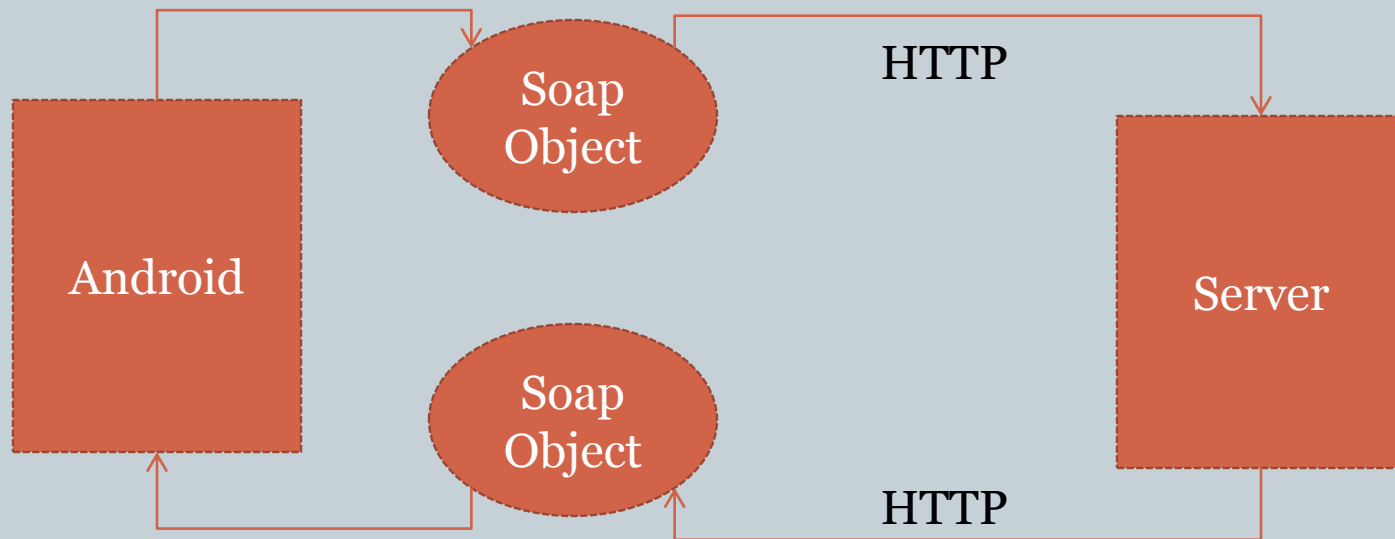
```
/**  
 * Goi webservice khi click vào button  
 */  
callServicesButton.setOnClickListener(new OnClickListener() {  
  
    @Override  
    public void onClick(View view) {
```

# III. Gọi và sử dụng web services từ android



- Sử dụng web services

- Việc truyền dữ liệu giữa server client sẽ thông qua **Http Protocol** và các **Soap Object**. Để có thể xử lý đối tượng SOAP ở trên android chúng ta sử dụng thư viện **Ksoap2** (Có đính kèm trong thư mục chứa mã nguồn)



# III. Gọi và sử dụng web services từ android



- Sử dụng web services

- Tạo **SoapObject** để gửi tham số lên server:

```
/**
 * Gọi web services
 */
String SOAP_ACTION = NAMESPACE + METHOD_NAME;
/**
 * Tạo SoapObject để chứa tham số gửi lên
 */
SoapObject soapRequestObject = new SoapObject(NAMESPACE, METHOD_NAME);
// Truyền tham số vào dưới dạng 1 property của Soap
soapRequestObject.addProperty("arg0", nameEditText.getText().toString());
SoapSerializationEnvelope envelope = new SoapSerializationEnvelope(SoapEnvelope.VERSION1);
envelope.setOutputSoapObject(soapRequestObject);
```

- Note: tên của tham số truyền lên phải xem trong file đặc tả wsdl để biết, nếu để eclipse tự động gen thì tên tham số sẽ theo thứ tự là “arg0” “arg1” ...v.v tùy vào lượng tham số của một method



# III. Gọi và sử dụng web services từ android



- Sử dụng web services:
  - Sử dụng Http protocol để gọi web services

```
String SOAP_ACTION = NAMESPACE + METHOD_NAME;  
// Truyền tham số vào dưới dạng 1 property của Soap  
soapRequestObject.addProperty("arg0", nameEditText.getText().toString());  
SoapSerializationEnvelope envelope = new SoapSerializationEnvelope(SoapEnvelope.VER11);  
envelope.setOutputSoapObject(soapRequestObject);  
  
// Dùng Http protocol để truyền tải dữ liệu qua lại giữa server và client  
HttpTransportSE androidHttpTransport = new HttpTransportSE(ENDPOINT_URL);  
try {  
    androidHttpTransport.call(SOAP_ACTION, envelope);  
} catch (IOException e) {  
    e.printStackTrace();  
} catch (XmlPullParserException e) {  
    e.printStackTrace();  
}
```

# III. Gọi và sử dụng web services từ android



- Sử dụng web services:
  - Phân tích dữ liệu từ đối tượng Soap trả về và cập nhật giao diện

```
/**
 * Dùng một Soap objects khác để chứa dữ liệu trả về
 */
SoapObject result = (SoapObject)envelope.bodyIn;

//Tùy vào đối tượng SOAP trả về của bạn phức tạp như thế nào mà phân tích nó để lấy dữ liệu cần thiết
String returnValues = result.getProperty(0).toString();

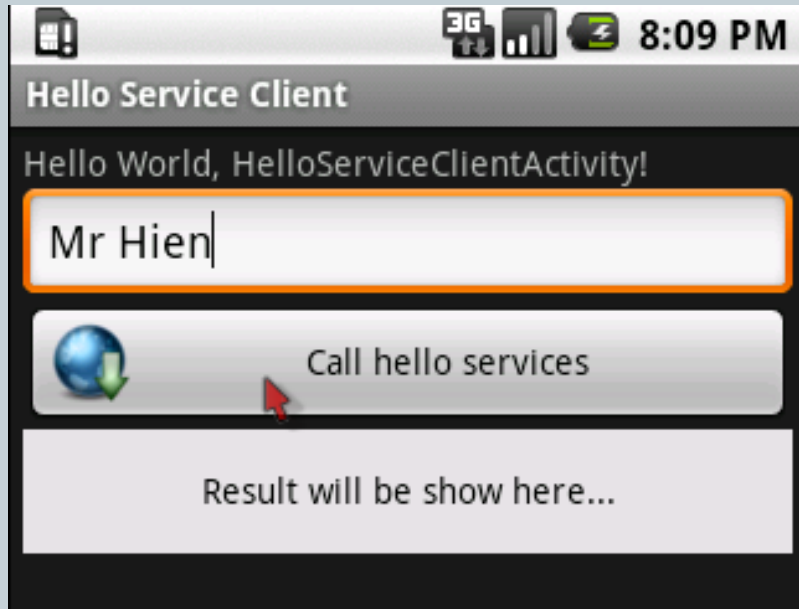
//Đổ dữ liệu vào các thành phần giao diện hiển thị lên màn hình
resultTextView.setText(returnValues);
```

- Note: Nếu đối tượng trả về phức tạp (ví dụ một Object hay là một List Object thì nên debug để xem cấu trúc của đối tượng SOAP trả về trước khi tiến hành bóc tách dữ liệu)

# III. Gọi và sử dụng web services từ android



- Demo
  - Chạy ứng dụng, nhập tên vào text box:



# III. Gọi và sử dụng web services từ android



- Demo

- Kết quả thu được sau khi nhấn call hello services:

