How to get the webservice client address for a jax-ws service depends on whether you are:

- Running your webservice as a servlet (in a Java EE container), or
- Running your webservice as a stand-alone application (Java SE 6 or 7).

Servlet Webservices If your webservice is a servlet then use the solution of the first post that contains the following:

```
HttpServletRequest req = (HttpServletRequest)mc.get(MessageContext.SERVLET_REQUEST);
```

Application Webservices : JAX-WS 2.1 If you are using a Java application (Java SE) you have no servlet context, so the HttpServletRequest will be null. You need to use the method of the later post, the one that has the following line:

```
HttpExchange exchange = (HttpExchange)msgx.get(JAXWSProperties.HTTP_EXCHANGE);
```

Note: this only works with the JAX-WS 2.1 stack/reference implementation.

Application Webservices : JAX-WS 2.2

In JAX-WS 2.2 the value of JAXWSProperties.HTTP_EXCHANGE has changed from "com.sun.xml.ws.http.exchange" (the value it was in JAX-WS 2.1) to "com.sun.xml.internal.ws.http.exchange". That means that a call to

```
HttpExchange exchange = (HttpExchange)msgx.get(JAXWSProperties.HTTP_EXCHANGE);
InetSocketAddress remoteAddress = exchange.getRemoteAddress();
String remoteHost = remoteAddress.getHostName();
```

will return null in JAX-WS 2.2 and you'll get a NullPointerException on the second line, and more importantly, cannot get the remote address of the client.

If you use the following call instead (using the literal value, ugh!):

```
HttpExchange exchange = (HttpExchange)msgx.get("com.sun.xml.ws.http.exchange");
InetSocketAddress remoteAddress = exchange.getRemoteAddress();
String remoteHost = remoteAddress.getHostName();
```

you will get a non-null value and will be able to obtain the client address.

So, how you get the remote address of the client depends on how you deploy your code (servlet or application) and which version of JAX-WS you are using (JAX-WS 2.1 or 2.2).

Recommendations

- **Servlets**: If you are deploying your JAX-WS webservice in a servlet you can always use the call to get the property MessageContext.SERVLET_REQUEST no matter what version of JAX-WS 2 you are using.

- **Applications**: If you are deploying your JAX-WS webservice in an application you can always use the call HttpExchange exchange = (HttpExchange)msgx.get("com.sun.xml.ws.http.exchange"); no matter whether you are using JAX-WS 2.1 or 2.2, therefore it is probably better to use the string literal in your code rather than the symbolic JAXWSProperties.HTTP_EXCHANGE.

As distasteful as using the literal is, it is better to have more robust code that works across JAX-WS versions rather than prettier code that doesn't.

I hope the JAX-WS team correct the issue sometime and restore the value of JAXWSProperties.HTTP_EXCHANGE to the useful value again.

Many thanks to the earlier posters that showed the various ways of finding the remote address of JAX-WS clients. The information was very useful :)