

Re: Ws-Addressing – WSE and vanilla Web Service Proxies

Source:

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.framework.webservices.enhancements/2006-07/>

- *From:* Steven L <StevenL@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx>
 - *Date:* Thu, 13 Jul 2006 05:59:02 -0700
-

William – great article :) With your help I managed to see how easy it is to get this working – by overriding the ProcessRequestMessage() in the router. Works fine.

However.... things are now working a little differently. The WSE will not work now as the address is given expects a SoapActor of the routing address at the target endpoint. I can't use this in the nonWSE client because "The <wsa:To> header must match the actor URI value of the web service."

So two questions.

(a) I can clearly get this working by figuring out when i have a WSE client (and delegate to the base) or a non_WSE client (and return a reference to the target endpoint). Is this good practice? Wuld i be better osmhoe determining WS-A compliant clients and delegating their calls?

(b) I am reading your paper over the course of the day, but in a nutshell, what am i missing in not calling the base of the ProcessRequestMessage()? It obviously does *something* because i get a "WS-A message header" type error when i leave it in. Does it simply parse the address and return the corresponding URI as defined in the referralCache (in which can i just need my own referralCache class) or does it do a bunch of other stuff?

Apologies if you have this in your paper – a lot of developers here are writing services on the assumption that my decision to use routing is a good one, so i want to make sure i'm not missing anything :)

cheers,
steven
<http://stevenR2.com>

"Softwaremaker" wrote:

Hi Steven,

My article was written to model a real-world system I was involved in :) We

Re: Ws-Addressing – WSE and vanilla Web Service Proxies

didn't have controls over (all) the desktops, which is why we needed the routing systems to inject goo into it ;) The principles are the same in a WSE3.0 or WCF world.

You will need minimally the wsa headers. Of course, I am assuming you are dealing with anonymous clients. You will have your hands full if you need to inject wss headers :)

Having said that – secured routing systems will need you to sign those wsa headers – so that is something you have to do as well if the policy calls for it.

--
Thank you.

Regards,
William Tay
<http://www.softwaremaker.net/blog>

=====

"Steven L" <StevenL@xxxxxxxxxxxxxxxxxxxxxxxxxxxx> wrote in message <news:9990E6BE-8C58-4F7E-9951-0B8CD11BB251@xxxxxxxxxxxxxxxxxxxx>

Hi William – that article is a good pointer, but looks primarily at routing

WSE 2.0 clients over a WSE 2.0 routing architecture.

I know the WSE client adds headers – it is the wsa Headers that need to be set by clients – that is the core point of what i need to know. I need to know what these are as other clients (non-WSE) will NEED to set them. I understand the architecture needs them, but that doesn't help me in knowing what to set.

Once the correct headers are sent to the WSE routing service i wouldn't have

to DO anything else (this is WSE framework stuff now) as it works with WSE proxy generated clients *without doing anything else*.

I will parse the article to see if it has some details on what headers need

to be set, but i'd love to see a real world article where all of your clients aren't all under your control (which I suspect is the majority of

the

planet!).

Thanks for the article pointer though – if you have any other related articles in your favourites, please let me know.

regards,
steven
<http://stevenR2.com>

"Softwaremaker" wrote:

Pardon me if I am wrong in understanding the requirements.

Like what Pablo had said, the proxies inherits differently from vanilla asp.net web services and those proxies injects headers into the messages transparent to you. It seems like the Routing service (hop-2-hop

endpoint

scenario) is expecting a s:Message with s:Headers (wsa Headers to be specific. For example, it needs to know where the s:FinalDestination

is).

Ultimately, the s:Intermediary SHOULD remove all processed headers

before

forwarding it to the next node so the s:FinalDestination has no part to

play

here.

In other words, you are trying to make use of a routing architecture

(which

is what SOAP is all about actually ... that POX/REST cannot do well)

without

Re: Ws-Addressing – WSE and vanilla Web Service Proxies

the required headers. You may need to introduce any layer of abstraction/indirection to make this work. You would have to inject

those

wsa goo into your s:Message one way or another for the Router to know

what

you intend to do. In other words, you may need to write your own wsa

headers

to the s:Message before it touches the wire. SOAP Extensions should be

able

to do that. It would be a lot of work for you if need to sign them as

well

;)

<http://msdn.microsoft.com/webservices/default.aspx?pull=/library/en-us/dnwse/html/securesoapnode.asp>

hth.

—
Thank you.

Regards,
William Tay
<http://www.softwaremaker.net/blog>

=====

"Steven L" <StevenL@xxxxxxxxxxxxxxxxxxxxxxxxxxxx>
wrote in message
<news:9F43B081-2777-4434-AA22-A750428C677B@xxxxxxxxxxxxxxxxxxxx>

Hi William – either you are or I am :)

Re: Ws-Addressing – WSE and vanilla Web Service Proxies

I do have a single end point – the problem is that I get a "Message Information Header Required" when i call the routing service from a non-WSE client

proxy.

All i want to know is how to get round this. Are you saying i need to

do

something on the server rather than set one or two headers on the

client?

You mention secure/insecure – why is that? I simply have a routing

service

and haven't moved to attach any security to it – a very basic web

service

end

point which i am simply accessing from clients via an intermediate WSE routing configuration.

I actually implemented exactly as stated in the WSE patterns book ...

but

unfortunately they didn't details how to actually *use* WS-Addressing

for

non-WSE clients.

steven
<http://stevenR2.com>

Re: Ws-Addressing – WSE and vanilla Web Service Proxies

"Softwaremaker" wrote:

Why not let your routing service (I assume it probably implements a

single

routing interface: ProcessMessage(m) and has a routing table as

well)

check

the contents (Content-based) for the necessary s:headers and route

it to

the

endpoint that can process it ?

In other words, you will have the same service (sort of) using 2

different

endpoints (one with security, one without) and they both can handle

the

specific implementations that are required. You may want to deploy

the

non-secured ones internally and the secured ones facing the cloud,

for

example. This is akin to the address/bindings/contracts model WCF

deploys.

It is difficult and I dont see the point of having a single service

endpoint

being able to handle secured and unsecured messages. Seems like an afterthought to me. What is the motivation of securing it in the

first

place

?

Am I missing something ?

--

Thank you.

Regards,
William Tay
<http://www.softwaremaker.net/blog>

=====

"Steven L"
<StevenL@xxxxxxxxxxxxxxxxxxxxxxxxxxxx>
wrote in message
news:724973C8-600D-4FD2-AD76-0FF5F814645B@xxxxxxxxxxxxxxxxxxxx

Gracias
Pablo.

The
pipeline
process is
something
i'm

Re: Ws-Addressing – WSE and vanilla Web Service Proxies

reasonable
familiar
with –

didn't

realise there
was such a
significant
difference
in the
generated

proxies.

Here is crux
of my
problem – i
need to
allow a
normal .Net
service

to

use

the routing
services i
am creating
– that is
essential
and some

may

not

use

WSE.

Got any
pointers on
making this
work?

saludos,
steven.
<http://stevenR2.com>

"Pablo
Cibraro"
wrote:

Hi
Steven,

The
proxies
created
by
WSE
are
completely
different
to
those

created

by

..NET.
The
WSE
proxies
intercept
the
SOAP
messages
and
execute
a

pipeline

where

the

message
is
transformed
to
a
different
version
(The
new

message

version

contains
WS-Addressing
and
WS-Security
headers).
The
WS-Addressing
headers
are
always
added
in
that
pipeline,
but

the

WS-Security
headers
are
not
(that
depend
on
some
WSE
specific

configuration,

the
WSE
policies).

The
same
happens
on
the
server
side,
if
you
configure
WSE,
a

WSE

pipeline

will
run
before
calling
to
your
service
(This
pipeline
performs

different

security
validations
and
removes
the
WS-Addressing
and

WS-Security

headers).

The
best
way
to
know
what
headers
are
required
is
to
enable
WSE

tracing

and

see
the
different

Re: Ws-Addressing – WSE and vanilla Web Service Proxies

messages
between
the
client
and
the
service.

Regards,
Pablo
Cibraro
<http://weblogs.asp.net/cibrax>
[MVP
–
Connected
Systems
Developer]

"Steven
L"
<Steven
L@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx>
wrote
in
message
<news:CEC76416-7CEC-464F-950C-A502B33EF494@xxx>

I
have
a
hub
performing
some
internal
routing
for
a
set
of

services

we

have,
using
the
WS-Addressing
support
in

Re: Ws-Addressing – WSE and vanilla Web Service Proxies

WSE
3.0.

When
a
vanilla
.Net
generated
proxy
client
tries
to
call
the

service

you

get:
"Microsoft.Web.Services3.Addressing.AddressingFaultMessage

Information

Header
Required"

When
a
WSE
generated
proxy
client
does
the
same
it
works

fine.

Can
someone
explain
or
point
me
at
the

different
in
the
client

proxy

configurations
between
these.

I
will
have
some
vanilla
.Net
clients
and
some
Java
clients

and so

i

need

to
tell
them
exactly
what
(Soap
Headers?)
to
set
on
their
proxy

client so

they
can
successfully
call
the
routing
service

–
i
don't
need
to

set

these

when

calling
the
service
directly,
so
any
detail
on
exactly
what

causes

this

requirement
by
the
WSE
would
be
greatly
appreciated!

Regards,
Steven
<http://stevenR2.com>

