Understanding Web Services Policy

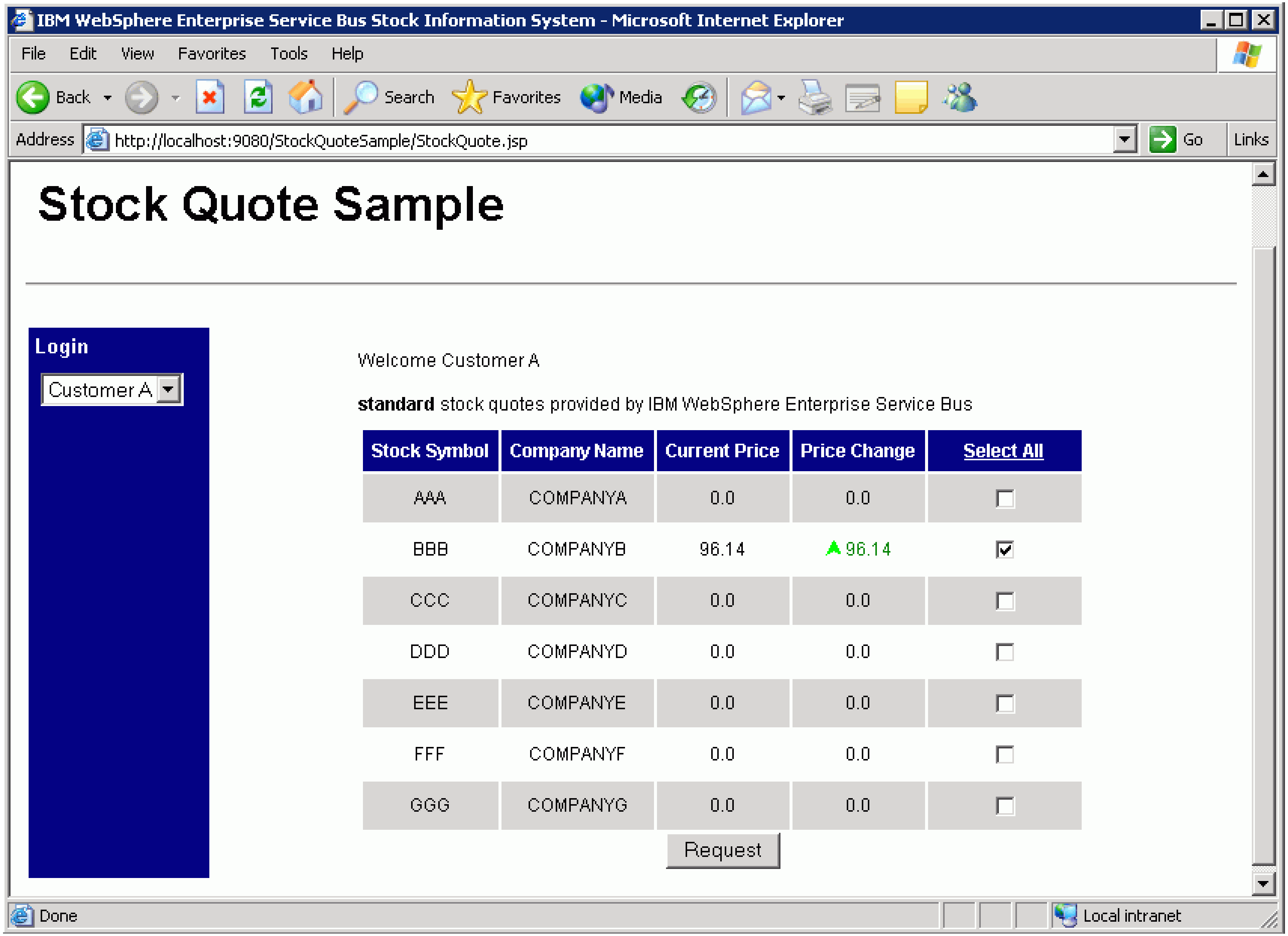
By
Prashanth Kamle
(08305006)

Characters

Tony

Acme Inc.

Tony's goal



IBM WebSphere Enterprise Service Bus Stock Information System - Microsoft Internet Explorer

http://localhost:9080/StockQuoteSample/StockQuote.jsp

Stock Quote Sample

Login

Customer A

Welcome Customer A

standard stock quotes provided by IBM WebSphere Enterprise Service Bus

Stock Symbol	Company Name	Current Price	Price Change	Select All
AAA	COMPANYA	0.0	0.0	□
BBB	COMPANYB	96.14	▲ 96.14	☑
CCC	COMPANYC	0.0	0.0	□
DDD	COMPANYD	0.0	0.0	□
EEE	COMPANYE	0.0	0.0	□
FFF	COMPANYF	0.0	0.0	□
GGG	COMPANYG	0.0	0.0	□

Request

WS Policy

- Web Services Policy is a machine-readable language for representing the capabilities and requirements (policies) of a Web service.

WS policy offers mechanisms to...

- represent consistent combinations of capabilities and requirements
- determine the compatibility of policies
- name and reference policies
- associate policies with Web service metadata constructs such as service, endpoint and operation.

WS Policy language

- Elements

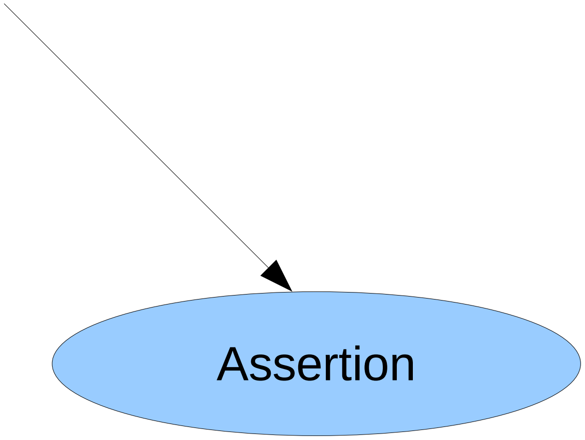
  - Policy
  - All
  - ExactlyOne
  - PolicyReference

- Attribute

  - wsp:optional

Policy Assertions

```
<Policy>
  <wsap:UsingAddressing />
</Policy>
```



Assertion

A Policy Assertion - Example

- Identifies the use of Web Services Addressing information headers.
- Client can recognize this policy assertion, engage addressing automatically.
- Uses `wsa:To` and `wsa:Action` in SOAP Envelopes.

Addressing - SOAP msg

```
<soap:Envelope>
  <soap:Header>
    <wsa:To>http://stock.acme.com/realquote</wsa:To>
    <wsa:Action>http://stock.acme.com/GetRealQuote</wsa:Action>
  </soap:Header>
  <soap:Body>...</soap:Body>
</soap:Envelope>
```

Policy Assertion – TL Security

```
<Policy>
  <wsap:UsingAddressing />
  <sp:TransportBinding>...</sp:TransportBinding>
</Policy>
```

Policy Assertion – TL Security

- Identifies the use of transport-level security – such as HTTPS
- Clients can recognize this policy assertion, engage transport-level security for protecting messages and include security timestamps in SOAP Envelopes

TL Security – SOAP msg

```
<soap:Envelope>
  <soap:Header>
    <wss:Security soap:mustUnderstand="1" >
      <wsu:Timestamp u:Id="_0">
        <wsu:Created>2006-01-19T02:49:53.914Z</u:Created>
        <wsu:Expires>2006-01-19T02:54:53.914Z</u:Expires>
      </wsu:Timestamp>
    </wss:Security>
    <wsa:To>http://real.acme.com/quote</wsa:To>
    <wsa:Action>http://real.acme.com/GetRealQuote</wsa:Action>
  </soap:Header>
  <soap:Body>...</soap:Body>
</soap:Envelope>
```

Hmmmmmm?

- What is a policy Assertion?
- What role does it play?

- a policy assertion is a piece of service metadata
- identifies a domain specific behavior that is a requirement (such as messaging, security, reliability and transaction)
- Used to convey a condition under which providers offer a Web service
- A policy-aware client can recognize policy assertions and engage these behaviors automatically

More Hmmmm?

- Who defines policy assertions?
- Where are they?

Hmmmmm......

- Defined by Web services developers, product designers, protocol authors and users.
- Like XML Schema libraries, policy assertions are a growing collection.

Hmmmmm....

- Several WS-* protocol specifications and applications define policy assertions

  - Web Services Security Policy
[http://schemas.xmlsoap.org/ws/2005/07/securitypolicy/]

  - Web Services Reliable Messaging Policy
[http://schemas.xmlsoap.org/ws/2005/02/rm/policy/]

  - Web Services Atomic Transaction
[http://schemas.xmlsoap.org/ws/2004/10/wsat/]

  - Web Services Business Activity Framework
[http://schemas.xmlsoap.org/ws/2004/10/wsba/]

  - Devices Profile for Web Services
[http://schemas.xmlsoap.org/ws/2006/02/devprof/]

Combining policy assertions

- Policy assertions can be combined in different ways to express consistent combinations of behaviors (capabilitiesand requirements).
- There are three policy operators for combining policy assertions:

```
    - Policy
    - All
    - ExactlyOne
```

- *addressing* and *transport-level security* are two policy assertions
- These assertions are combined using the `All` operator
- Combining policy assertions means that all the behaviors represented by these assertions are required

Addressing and Security Policy Expression

Case 1: Addressing and TL security

```
<All>
  <wsap:UsingAddressing />
  <sp:TransportBinding>...</sp:TransportBinding>
</All>
```

Addressing and Security Policy Expression

Case 2: Transport or Message-Level Security Policy Expression

```
<ExactlyOne>
   <sp:TransportBinding>...</sp:TransportBinding>
   <sp:AsymmetricBinding>...</sp:AsymmetricBinding >
</ExactlyOne>
```

Combined!

```
<All>
  <wsap:UsingAddressing />
  <ExactlyOne>
     <sp:TransportBinding>...</sp:TransportBinding>
     <sp:AsymmetricBinding>...</sp:AsymmetricBinding >
  </ExactlyOne>
</All>
```

Optional Policy Assertion

```
<All>
  <mtom:OptimizedMimeSerialization wsp:Optional="true"/>
  <wsap:UsingAddressing />
  <ExactlyOne>
     <sp:TransportBinding>...</sp:TransportBinding>
     <sp:AsymmetricBinding>...</sp:AsymmetricBinding >
  </ExactlyOne>
</All>
```

Optional Policy Assertion

- There are behaviors that may be engaged (in contrast to must be engaged) for a Web service interaction.
- A service provider will not fault if these behaviors are not engaged.
- A policy assertion is marked as optional using the `wsp:Optional` attribute
- Optional assertions represent the capabilities of the service provider as opposed to the requirements of the service provider.

Nested policy expressions, security as example

- Must know what transport token to use, what secure transport to use, what algorithm suite to use for performing cryptographic operations, etc
- A nested policy expression can be used to enumerate such dependent behaviours.
- A nested policy expression is a policy expression that is a child element of a policy assertion element.
- A nested policy expression further qualifies the behavior of its parent policy assertion.

A nested policy

```
<sp:TransportBinding>
  <Policy>
    <sp:TransportToken>
      <Policy>
        <sp:HttpsToken RequireClientCertificate="false" />
      </Policy>
    </sp:TransportToken>
    <sp:AlgorithmSuite>
      <Policy>
        <sp:Basic256Rsa15 />
      </Policy>
    </sp:AlgorithmSuite>
    …
  </Policy>
</sp:TransportBinding>
```

Nested policies...

- a policy-aware client that recognizes this policy assertion and engages transport-level security and its dependent behaviors automatically
- the complexity of security usage is absorbed by a policy-aware client and hidden from these Web service developers.

Referencing Policy Expressions

- A policy expression may be identified by a URI and referenced for re-use as a standalone policy or within another policy expression.
- `PolicyReference` element can be used to reference a policy expression.

Example

- Common Policy Expression

```
<Policy wsu:Id="common">
  <mtom:OptimizedMimeSerialization wsp:Optional="true"/>
  <wsap:UsingAddressing />
</Policy>
```

- *PolicyReference* to Common Policy Expression

```
<Policy wsu:Id="secure">
  <All>
    <PolicyReference URI="#common"/>
    <ExactlyOne>
      <sp:TransportBinding>...</sp:TransportBinding>
      <sp:AsymmetricBinding>...</sp:AsymmetricBinding >
    </ExactlyOne>
  </All>
</Policy>
```

Attaching Policy Expressions to WSDL

- The WSDL `binding` element is a common policy attachment point.

```
<wsdl:binding name="SecureBinding"
type="tns:RealTimeDataInterface" >

  <PolicyReference URI="#secure" />

  <wsdl:operation name="GetRealQuote" >...</wsdl:operation>

  ...

</wsdl:binding>
```

Conclusion

- Providers have the option to convey requirements through word-of-mouth and documentation
- Companies can preserve and leverage their investments in WSDL and represent the capabilities and requirements of a Web service as policies.
- Policy-aware clients understand policy expressions and engage the behaviors represented by providers automatically.
- Web Services Policy extends the foundation on which to build interoperable Web services, hides complexity from developers and automates Web service interactions.

References

- http://msdn.microsoft.com/en-us/library/ms996497.aspx
- http://www.w3.org/Submission/WS-Policy/

?