

EntwicklerCamp March, 2010

Bob Balaban
Looseleaf Software
bbalaban@gmail.com

REAL-WORLD WEB SERVICES: TECHNOLOGIES AND WORKAROUNDS



Copyright 2010 Looseleaf Software

Agenda

- Speaker introduction
- Deconstructing "Web Services"
 - Origins and building blocks
 - Kinds of Web Services
 - Design time technologies
 - Run time technologies
- What Can Go Wrong? (you have no idea)
- Workarounds? (sometimes)
- Summary
- Q & A

Copyright 2010 Looseleaf Software

Speaker Introduction

- 15+ years as a Lotus/Iris/IBM developer (1-2-3, Notes/Domino)
 - Wrote the original LotusScript/Java back-end classes (Notes v4.0)
 - Notes/SAP integration (v7.x, 2006)
- 9+ years as an independent consultant/developer/author
 - Frequent speaker at LUGs and Lotusphere
 - Many trade journal articles
 - Expertise in N/D programmability, plugins, J2ee, system integration, messaging, appdev in general

Copyright 2010 Looseleaf Software

Origins of Web Services

- The "Web" evolved from static, hyperlinked text pages:
- First to dynamic queries
- Then to interactive form-based processing
- But, always optimized for a Human at the browser
- B2B, B2C, B2B2C.....
- What about A2A?
 - i.e., how do I get *programmatic* access to *stuff*?

Copyright 2010 Looseleaf Software

Origins -2

- Why not use the infrastructure of the Web for Application-initiated transactions?
- HTTP, SSL
- Same business logic and data stores
- What would we use for data transfer formats?
- XML!
- We can also create conventions for using XML to invoke COMMANDS, not just "queries"
- And thus – SOAP!

Copyright 2010 Looseleaf Software

Origins - 3 (SOAP)

- Web Services are analagous to RPC:
- "Client" initiates a parameterized request over a network to a "server"
- Server and target application are locatable over the network (DNS)
- Request has well-known ID (method name)
- Request has well-known parameters (primitives, objects...)
- Response has well-known structure

Copyright 2010 Looseleaf Software

So What? (SOAP)

- This architecture has several big advantages:
- Uses HTTP (and other!) transports
- XML "wire protocol" allows for vendor (and OS!) independence
- "Meta" standards for description and publishing allow for additional automation
- WSDL, UDDI
- Publish, discover

Copyright 2010 Looseleaf Software

And... ?

- So, Web Services can be thought of as Business Components
- Implementations of client (user) and server (provider) are no longer tightly-coupled!
- Invocation/response are message-based
- And the message structure is standardized

Copyright 2010 Looseleaf Software

And...?? (Still SOAP)

- XML - Data description
- SOAP - Simple Object Access Protocol
- XML request/response messaging
- WSDL - Web Services Description Language
- XML description of a service
- And the usual XML Parser suspects!

Note: There's nothing about this that is Java only!

Copyright 2010 Looseleaf Software

Demo:

Simple Java web service on Domino
with LotusScript and .NET clients
(ok, that's really 2 demos...)

Copyright 2010 Looseleaf Software

Other Kinds of Web Services?

- Almost anything can be a “service”
 - Client/server architecture
 - “SOA” (Service Oriented Architecture)
- BUT, two kinds are widespread “standards”:
 - SOAP (Simple Object Access Protocol)
 - REST (Representational State Transformation)
 - Sounds like a Ph.D. dissertation? It was!
- Note: Web Services is a TOOL, not a RELIGION!

Copyright 2010 Looseleaf Software

Short Description of REST

- Treats server resources as URLs
 - Returned “page” is either data (if URL is a node in the tree), or a collection of links
 - Very similar to existing Domino ?ReadViewEntries commands
 - “Atom feeds” are often used to organize lists of entries (IETF rfc #4287)
 - <http://www.ietf.org/rfc/rfc4287.txt>
- Great book: “Restful Web Services”, by Sam Ruby and Leonard Richardson, pub. O’Reilly

Copyright 2010 Looseleaf Software

Examples of REST APIs/Web Services

- Google:
 - Google Apps supports various libraries of Atom-oriented RESTful services to mail, calendar, docs, sites, etc. etc.
 - Because Atom is not so easy to deal with, they also provide client-side object libraries to make it easy (Java, c#, JavaScript, PHP, etc.)
- Domino 8.5.2 (coming!)
 - Not public yet, can't talk about it... ☹

Copyright 2010 Looseleaf Software

Common Building Blocks of Web Services

- XML – in various forms
 - XSD (schema)
 - WSDL (Web Services Description Language)
 - OOXML (Object-oriented XML)
 - And off-the-shelf tools!
- HTTP – send and receive XML between clients and servers

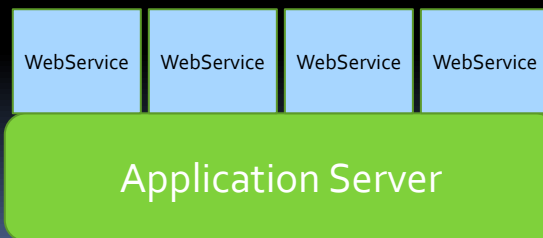
Copyright 2010 Looseleaf Software

So, How Does It Work?

- REST is pretty easy to understand
 - Formulate a URI to a resource: a piece of data (e.g., document) or collection (e.g., view)
 - Use HTTP GET to fetch data/links
 - Use HTTP POST to store data
 - Content format is usually Atom, but could be anything (e.g., DXL!)
- SOAP is more complicated
 - Requires technology at both DesignTime AND RunTime

Copyright 2010 Looseleaf Software

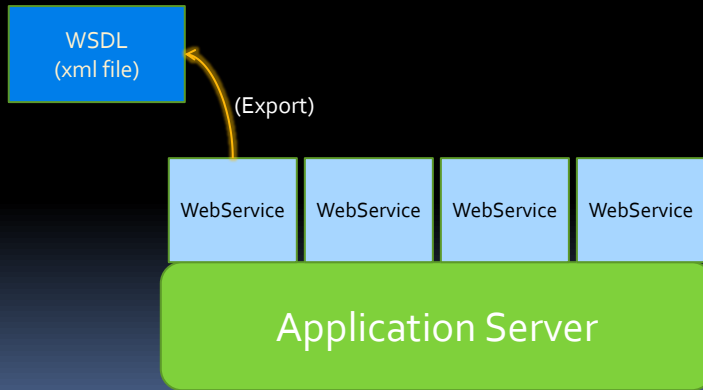
SOAP Service Implementation Process



WAS, Domino, etc.

Copyright 2010 Looseleaf Software

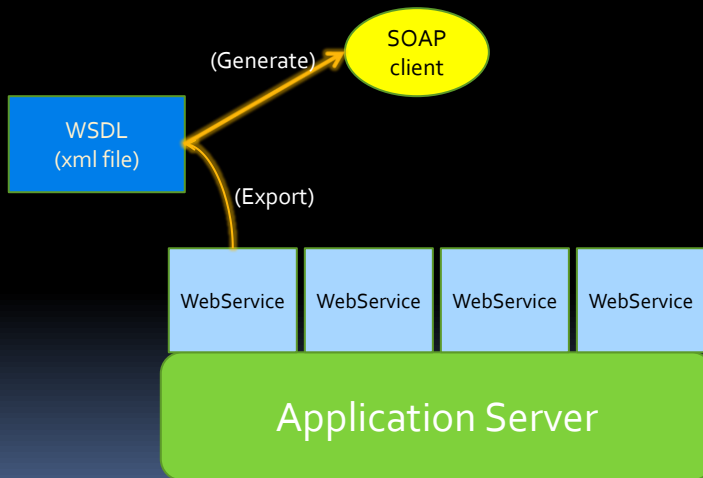
SOAP Service Implementation Process



WAS, Domino, etc.

Copyright 2010 Looseleaf Software

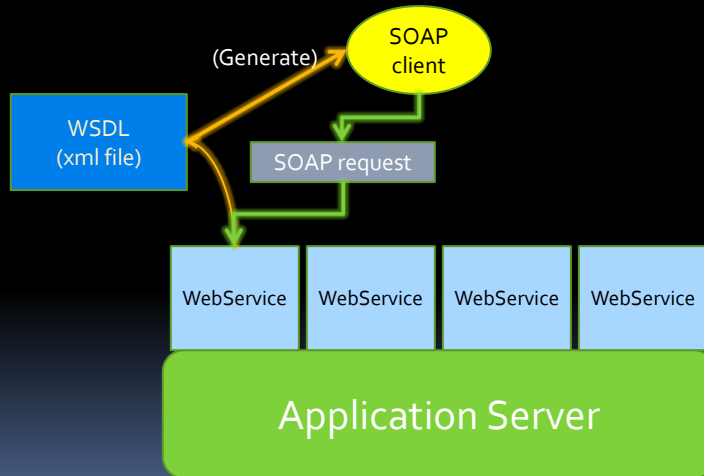
SOAP Service Implementation Process



WAS, Domino, etc.

Copyright 2010 Looseleaf Software

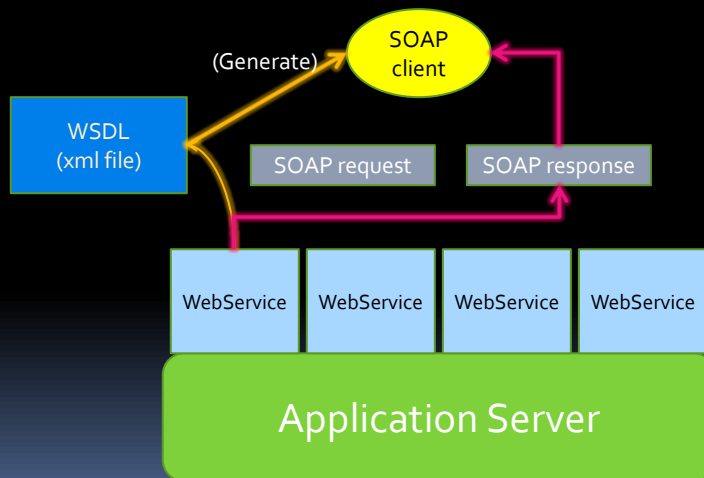
SOAP Service Implementation Process



WAS, Domino, etc.

Copyright 2010 Looseleaf Software

SOAP Service Implementation Process



WAS, Domino, etc.

Copyright 2010 Looseleaf Software

DesignTime Technologies

- Two choices for building the provider:
 - Generate WSDL from implementation class
 - Write the code/interface first, then create WSDL
 - Generate implementation "skeleton" from existing WSDL (also works for WS consumer!)
- WSDL must include: location URI, interface description, custom object definitions (results, arguments)

See "ListObject" web service demo

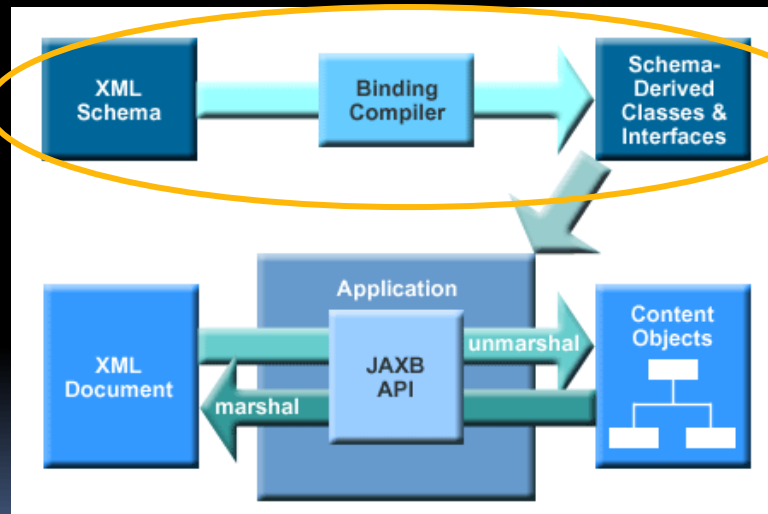
Copyright 2010 Looseleaf Software

How to Map WSDL/Code?

- Q: Given code, how do you generate WSDL?
- Q: Given WSDL, how do you generate code?
- A: Use the free Java (or other) tools!
 - Java: Apache AXIS (<http://ws.apache.org/axis2>)
 - Sun JAXB (Java Architecture for XML Binding)
 - .NET: Visual Studio
 - LotusScript: Domino Designer
- Note: Designer uses AXIS for Java web services in Domino

Copyright 2010 Looseleaf Software

How DesignTime Mapping Works



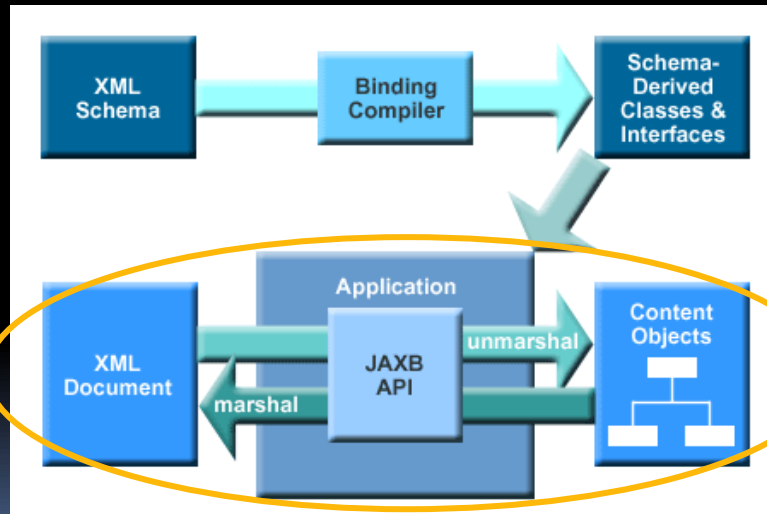
<http://java.sun.com/developer/technicalArticles/WebServices/jaxb/>
Copyright 2010 Looseleaf Software

DesignTime Work

- Given WSDL... (or WSDL + XSD)
 - Create object implementations for each defined "type"
 - Objects have no real behaviors, just slots for data elements (like structs in C, or JavaBeans)
 - Generate "controller" code to handle mapping incoming XML "objects" to target language "objects" (LS, Java, c#...)
 - More controller code to instantiate the service class (developer has to write service implementation) and invoke it with input objects (if any) "unmarshall"
 - Generate Response code to take returned values/objects and serialize to XML – "marshal"

Copyright 2010 Looseleaf Software

RunTime Technologies



Copyright 2010 Looseleaf Software

Note!

- The *tools* are NOT standard
- However, the XML they produce that is sent over the wire IS standard
- So, AXIS and JAX-B (and Visual Studio...) may not work exactly the same way
 - We will see an example of this soon...

Copyright 2010 Looseleaf Software

RunTime Tasks

- Not too many, all the code is already generated at DesignTime
- AppServer framework receives HTTP POST call specifying target Web Service in the URL, with the input "payload" attached
- Locate service "handler" object previously registered (at deployment time) with the framework, invoke it
 - Inputs are unmarshalled to objects, service implementation is invoked
- Format SOAP response envelope from unmarshalled (serialized) response object and return to client

Copyright 2010 Looseleaf Software

Sample SOAP Message

```
POST /StockQuote HTTP/1.1
Host: www.stockquoteserver.com
Content-Type: text/xml; charset="utf-8"
Content-Length: nnnn
SOAPAction: "Some-URI"
<SOAP-ENV:Envelope
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
<SOAP-ENV:Body>
<m:GetLastTradePrice xmlns:m="Some-URI">
<symbol>DIS</symbol>
</m:GetLastTradePrice>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Copyright 2010 Looseleaf Software

Sample SOAP Response

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset="utf-8"
Content-Length: nnnn
<SOAP-ENV:Envelope
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
<SOAP-ENV:Body>
<m:GetLastTradePriceResponse xmlns:m="Some-URI">
<Price>34.5</Price>
</m:GetLastTradePriceResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

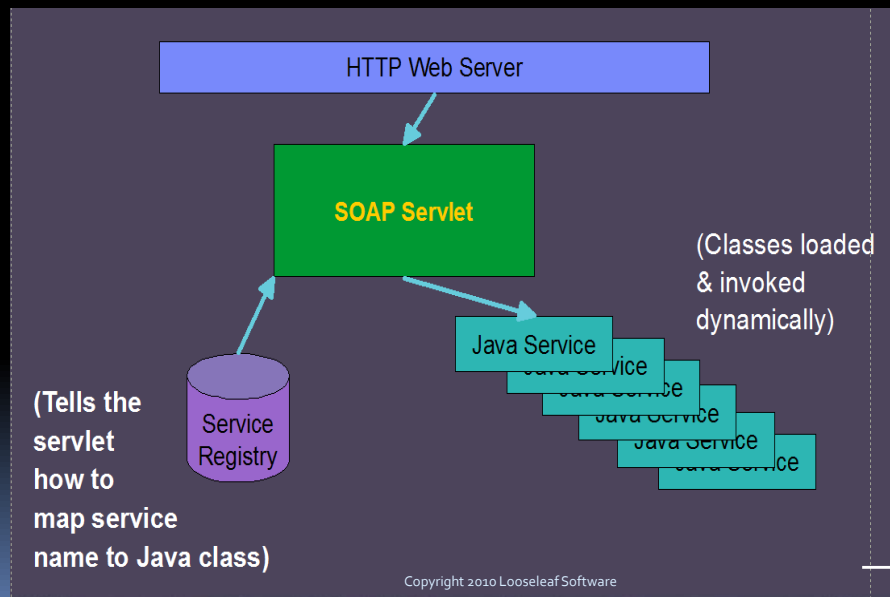
Copyright 2010 Looseleaf Software

Developer View?

- Get an instance of the "service" object
- Invoke method(s) on it
- Get responses back
- It will even throw exceptions!
 - Special SOAP response label called "FAULT" can be returned instead of an expected value/object.
 - Client code in target language raises the exception when it sees this
- So you should always use try/catch (or language equivalent)

Copyright 2010 Looseleaf Software

Another Picture (J2ee)



Summary of SOAP Benefits

- Vendor neutral
- Implementation language neutral
- Framework neutral
- Standards based (HTTP, XML)
- Handles simple and complex data types well
- Free off-the-shelf tools available
- Enhanced developer productivity

So What Could Possibly Go Wrong?

- Believe it or not...
- There are a few things that can go wrong:
 - Spec lawyers
 - Security (huh?)
 - Character sets
 - “Why would you want to do THAT?”
- Let’s look at each of these, with some more examples...

Copyright 2010 Looseleaf Software

Wrong: Spec Lawyers

- Something to try:
 - Get Rational App Developer (there’s a free trial available)
 - Sign up for a free developer ID with Amazon Web Services
 - Tell RAD to generate a client for the Amazon WSDL
 - Write a nice note (PMR will do too) to IBM asking why you get > 100 errors for something so simple

Copyright 2010 Looseleaf Software

Spec Lawyers - 2

- Why does this happen?
- Amazon.com WS are implemented in .NET
- RAD (of course) is J2EE
- Amazon has several service interfaces that define arrays of objects
 - .NET supports this very well
 - J2EE WS do not
- IBM and Microsoft (who jointly developed SOAP in the first place) do not agree on whether arrays of objects are "standard" or not
- It's a spec loophole
- Workaround: use a container object for the list/array (as with ListObject demo)

Copyright 2010 Looseleaf Software

Wrong: Security (what?)

- What security?
- "Optimistic security" == "No security"
 - Let's think about what you would need:
 - Message reliability (did it get there, or not?)
 - Make the XML traffic unreadable (encrypt it)
 - Selectively encrypt message (e.g., payload only, not routing info)
 - Authenticate requests on server
 - Authenticate responses on client
 - Specify access control to services on server

Copyright 2010 Looseleaf Software

Security - 2

- What do we have "accepted" solutions for?
 - Meaning: standards, plus implementations
- Make the XML traffic unreadable (encrypt it)
 - SSL (easy)
 - Or other key-exchange (easy)
- Authenticate requests on server
 - Requires some kind of digital signature
 - Which requires shared certificate chains
 - And trusted directory
 - And PKCS infrastructure

Copyright 2010 Looseleaf Software

Security - 3

- Authenticate responses on client
 - Ditto all
- Specify access control to services on server
 - Requires authenticated user (as above)
 - And a trusted directory service
- In short, nothing has quite made it happen yet
- Therefore, most implementations are for:
 - Trivial, public data (weather, phone numbers, zip codes...)
 - On secure intranets

Copyright 2010 Looseleaf Software

Security - 4

- Multi-hop SOAP
 - Client->Server->Server->Server
 - Credentials, signature, etc. must follow all the way
- Agreement on order of operations
 - Sign first, then encrypt?
 - Encrypt first, then sign?
- Is there a way to do ACL in a cross-product fashion?
 - Perhaps not. Directories will rule!

Copyright 2010 Looseleaf Software

Security - 5

- If you're using all IBM platforms you get more choices
- RAD lets you create Web Services (and clients) with WAS-only security options
- XML digital signatures
- XML encryption
- NOTE: You have to pick the SAME security option for BOTH client and service!

Copyright 2010 Looseleaf Software

Wrong: Character Sets

- If everyone uses UTF-8 for both clients and servers, no problem!
- BUT!
 - Not everyone does ☹
 - ISO-8859-1, ISO-8859-8
 - Chinese (traditional or simplified)
 - SHF-JIS
 - UTF-16?? (less of a problem, easily converted to utf-8 and back)
- Should translation burden be in the infrastructure, or the applications?
- Workaround: standardize on Unicode

Copyright 2010 Looseleaf Software

Wrong: “Why Would You Want to do THAT?”

- Knowing the basics of the architecture and tooling, you want to apply it to a real-world situation – seems logical!
 - BUT, you happen to step outside the boundaries of what the system “supports”!
 - E.g., you want to return an array of objects from the WS (a list of customer records, maybe)
 - Works in Visual Studio!
 - Does not work in J2EE!
 - Differing interpretations of the SOAP spec!

Copyright 2010 Looseleaf Software

WWYWTDT? – In-depth Example

- Many Web Services are front-ends to pass queries to a relational dbms
 - WS uses (e.g.) JDBC on server to access Tier-3 dbms
 - WS converts result set to something the WS can return to the client (container object, etc.)

Demo: Simple JDBC

Copyright 2010 Looseleaf Software

So Far, So Good!

- Now the customer says: “I want the WS to return a `javax.sql.rowset.CachedRowSet` object”
 - Uh oh.
- First, does the Domino version we have include a JRE that contains this class?
 - Yes (8.5.x does)
- Look up the javadoc: Uh oh.
 - It's not a class, it's an Interface
 - WS do not do well when you try to return an Interface

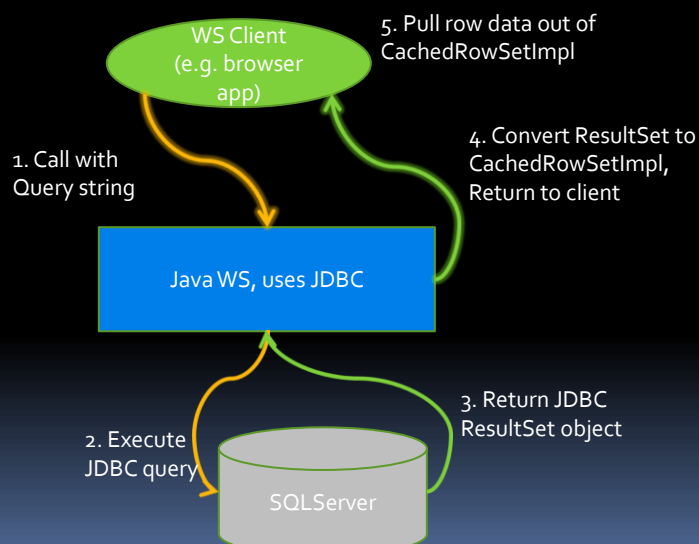
Copyright 2010 Looseleaf Software

Speedbump!

- So, we need to find a class that *implements* `javax.sql.rowset.CachedRowSet`, then we can (maybe) return that as a container object
- There is one!
`com.sun.rowset.CachedRowSetImpl`
 - (AND, you can find the source code on the Web!)
- `CachedRowSetImpl` can be built from a JDBC `ResultSet`!
- So, let's just build a Domino Java WS that returns this object!

Copyright 2010 Looseleaf Software

Architecture of Our Query WS



Copyright 2010 Looseleaf Software

Speedbump!

- Need SqlServer JDBC driver available in Designer AND on server!
- Download Microsoft driver (free!)
 - www.microsoft.com/msdn, search for "JDBC driver" → sqljdbc.jar
 - Works with SqlServer Express v8 (also free!)
 - If you have Db2, it's basically the same thing
- Copy to Domino\jvm\lib\ext and Notes\jvm\lib\ext
- Do NOT waste time trying to use JavaUserClasses in notes.ini!

Copyright 2010 Looseleaf Software

Demo: Simple JDBC WS

- First, let's write a simple WS using JDBC, then we can get fancy with it

Demo: SimpleJDBC WS

Copyright 2010 Looseleaf Software

Demo: Uh Oh!!

- Another Speedbump!

```
03/02/2010 12:19:20 PM HTTP Server: Started
03/02/2010 12:19:25 PM HTTP JVM: java.lang.SecurityException: not allowed to make a socket connection to localhost,1433
03/02/2010 12:19:25 PM HTTP JVM: at lotus.notes.AgentSecurityManager.checkConnect(AgentSecurityManager.java:233)
03/02/2010 12:19:25 PM HTTP JVM: at com.microsoft.sqlserver.jdbc.SQLServerConnectionSecurityManager.checkConnect(Unknown Source)
03/02/2010 12:19:25 PM HTTP JVM: at com.microsoft.sqlserver.jdbc.SQLServerConnection.connect(Unknown Source)
03/02/2010 12:19:25 PM HTTP JVM: at com.microsoft.sqlserver.jdbc.SQLServerDriver.connect(Unknown Source)
03/02/2010 12:19:25 PM HTTP JVM: at Untitled.GetResults(Unknown Source)
```

Workaround: set "unrestricted" on web service security
Sometimes you have to edit `\jvm\lib\security\java.policy`

Copyright 2010 Looseleaf Software

Next Step (Still WWYWTDT?)

- Can we have the Web Service return an actual `com.sun.rowset.CachedRowSetImpl` instance?
 - I spent 3 weeks on this question for a customer
 - So I can save YOU LOTS of time. The answer is...
 - NO
- Domino will refuse to generate the WSDL, because the `CachedRowSet` interface is in the `javaX` package (AXIS won't handle it)
- However, RAD and WAS will (sort of) do it! (They use JAX-B)

Copyright 2010 Looseleaf Software

Demo: CachedRowSetImpl and RAD/WAS

- It only sort-of works:
 - Generated classes are not com.sun or even javax.sql.rowset
 - They are in your "default namespace" package (e.g., "looseleaf" or whatever you declare)
- So, even having the real implementations available (on client AND server), doesn't help
- You will essentially have to re-implement the functionality
- Because SOAP is not good at object functionality, only at data packaging

Copyright 2010 Looseleaf Software

One Final Topic (if there's time)

- Connection pooling!
 - It's a VERY good thing
- Creating a connection from your WS to a data resource (NSF, RDB, anything) is usually a bit expensive (you have to log in...)
- Instead of connecting/disconnecting on every transaction, why not keep a bunch of "connection" objects around?
- Get one from the pool, it's already "open"
- If there are none, get one the normal way
- Connection "close" becomes a no-op

Copyright 2010 Looseleaf Software

JDBC Supports ConnPools

- Instead of loading a “driver” and using it to create a connection,
- You attach to a pooled “Data Source”, pooling is handled automatically
 - Whooo hoo!
- EXCEPT...
 - It isn't JDBC that manages the pool
 - The data source has to be created on the AppServer
- WAS can do it ☺
- Domino cannot ☹
- Or you can write your own (make sure it's persistent!)

Copyright 2010 Looseleaf Software

Web Services Summary

- GREAT functionalty, as far as it goes
 - Standards based
 - Solves real problems
 - Tools/IDEs are mostly good and pretty easy to use
- You have to be VERY careful to not step outside the limits of what the technology does well
 - If you do, you are OYO (on your own)
- SOAP and REST are both heavily oriented towards DATA transfer, not so much BEHAVIOR transfer (objects with real code behind them)

Copyright 2010 Looseleaf Software

Danke Schoen!

Q & A

bbalaban@gmail.com

Blog: <http://www.bobzblog.com>

Copyright 2010 Looseleaf Software