Towards a Common Java LDAP API

Emmanuel Lecharny
Apache Directory
elecharny@apache.org

Ludovic Poitou
OpenDS
ludovic.poitou@sun.com
A poor man's choice:

JNDI
or
JLDAP/Netscape LDAP SDK
Good News:

Apache Directory Client API
OpenDS Client SDK
UnboundID LDAP SDK
Why this resurgence of activity?

- Servers need to issue outbound connections
- A directory product is wrapped with tools
- Previous effort to agree on standard API stalled
Is the LDAP developers ecosystem that large?

We just need one good common API
The foundation

- JLDAP/Netscape SDK as template
- Leveraging Java5 language constructions
- Synchronous and Asynchronous methods
Connection
Request
Response
Handling Data
Connection

- LDAPConnection is an interface
- Factory
  - to hide I/O library
  - to handle single connection vs pooled connections
- Referral Policies (including Authentication)
Request / Responses

Synchronous vs Asynchronous
3 approaches
Just a question of style
Handling data

Entries
Attributes
Values
Names
Entry description

• Base object for the server
• Contains the DN and the Attributes
• Uses ellipsis:
  entry.add( "ObjectClass", "top", "person" );
• Constructor or factory?
Example : JNDI

DirContext ctx = new InitialDirContext( env );

Attributes attrs = new BasicAttributes( true ); // case-ignore
Attribute objclass = new BasicAttribute("objectclass");
objclass.add("top");
objclass.add("organizationalUnit");
attrs.put(objclass);
attrs.put( "ou", "fruits" );

Context result = ctx.createSubcontext("ou=Fruits", attrs);

Example : New API

Entry entry = new EntryImpl(  new DN( "ou=fruits, dc=example, dc=com" ) );

entry.add( "objectClass", "top", "organizationalUnit" );
entry.add( "ou", "fruits" );

connection.add( entry );
Values description

- Much more an internal representation of attribute data
- Should it be exposed to the client?
- It contains
  - either String (for H/R attributeTypes)
  - Or byte[] (Binary AttributeTypes)
- From the server point of view, everything is a byte[]
<table>
<thead>
<tr>
<th>Ldap Objects</th>
<th>Ldap Ghads</th>
<th>Ldap Glads</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Search responses

• Enumeration of Searched Entries
• Should search always try to page (like RDMBs queries)?
• Should it be additional APIs built on the basic Search request?
Schema

• Schema is required for handling DNs, Entries, Values...

• Some servers may return non schema compliant data

• Application may not know Server's schema or works with multiple servers with different schema

• API needs to handle all scenarii
Schema

• Default schema must be present
  – At least the standard one
• Should be able to extend from file or server
• How to deal with server specific schema elements requiring code (Syntax, MatchingRules) ?
Other aspects

• Security: TLS handled under the hood
• Exceptions
• Extended operations
• Controls
• DSML
  • Reader / Writer / Utilities
• LDIF
  • Reader / Writer / Utilities
Conclusion

• There is NO common API yet.
• There are many aspects to address as a community
• Server vendors have their needs and their proposed solutions
• Application developers should participate and provide input
• It will take time to have that single best of the world common LDAP JAVA API !
• Interested ? Join the effort !