oose.
Innovative Informatik

LDAP.
Eine Einführung für Java-Entwickler.

Berlin, 14.09.2008

berlin.jar
... die Java-Konferenz in Berlin

Stefan Zörner, oose Innovative Informatik GmbH

LDAP. Eine Einführung für Java-Entwickler

oose. Innovative Informatik

Stefan Zörner – Stationen

1991-94 **Ausbildung** Math.-techn. Assistent bei der Bayer AG
Studium Mathematik (Diplom 1998), Schwerpunkt Informatik
1998-2001 **Mummert + Partner** AG, Berater, u.a. Sun-Trainer
2001-2006 **IBM** e-business Innovation Center, IT-Architekt
Seit Juli 2006 :
Berater und Trainer bei **oose** Innovative Informatik GmbH
Stefan.Zoerner@oose.de

Veröffentlichungen, Vorträge (Auswahl)

Bücher „Portlets“, 2006
„LDAP für Java-Entwickler“, 3. Auflage 2007

Artikel in Java Magazin und bei IBM developerWorks
Vorträge bei JAX und W-JAX seit 2002, Advisory Board W-JAX

Sonstiges

Mitarbeit im **Apache Directory Project**, seit August 2005 als Committer, seit 2006 im PMC, szoerner@apache.org

OMG Certified UML Professional (Intermediate)
Sun Certified Web Component Developer for J2EE
IBM Certified Solution Developer - WebSphere Portal V5.1

Agenda

1. Warum sollte mich LDAP interessieren?
2. Was ist LDAP überhaupt? Ein paar (!) Details.
3. Wie spricht man zu LDAP mit Java?
4. Wie integriert man LDAP als Benutzerdatenbasis?
5. Wenn Sie neugierig geworden sind …

➔ Warum sollte mich LDAP interessieren?

- Verzeichnisse, Verzeichnisdienste
- LDAP
- Anwendungsgebiete

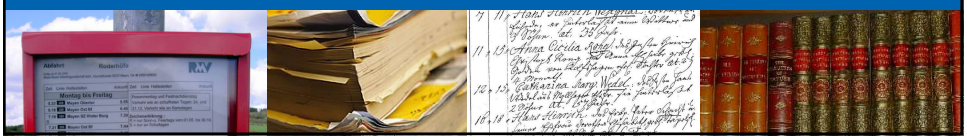
Der Begriff des Verzeichnisses in der realen Welt

Verzeichnis:

- Auflistung oder Sammlung von Informationen
- dient dem Zweck, Informationen zu bewahren und bei Bedarf Interessierten zugänglich zu machen

Beispiele für Verzeichnisse der realen Welt

- Telefonbücher (öffentliche oder unternehmensinterne)
- Fahrpläne
- Werksverzeichnisse (z.B. Köchelverzeichnis für Werke Mozarts)
- Kirchen- und Grundbücher



LDAP. Eine Einführung für Java-Entwickler
oose. Innovative Informatik

Der Begriff des Verzeichnisses in der Welt der Informationstechnologie

Begriff „Verzeichnis“ etabliert für :

- Spezieller Datenspeicher
- Speicherung der Daten erfolgt in Form so genannter Einträge
- Die Menge der Einträge bildet eine baumförmige Struktur (hierarchische Datenbank)

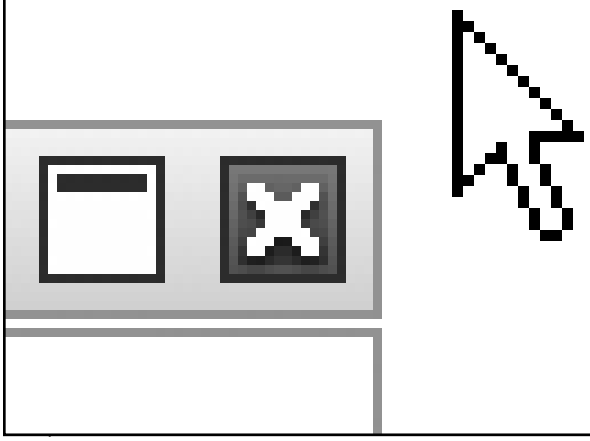
Verzeichnisdienst

- Lösung, die Nutzern den Zugang zu einem Verzeichnis ermöglicht (etwa um Informationen aus dem Verzeichnis abzurufen)
- Anschauliches Beispiel (reale Welt): Telefonauskunft
- Im EDV-Bereich in der Regel eine Softwarekomponente

LDAP – Lightweight Directory Access Protocol

- TCP/IP-basiertes Protokoll, um Operationen auf Verzeichnissen durchzuführen (z.B. Suchen, Anlegen und Ändern von Einträgen)

Demo: Zugriff auf ein Adressbuch mit Mozilla Thunderbird …



LDAP genügt dem Client/Server-Modell. Als Clients können sehr unterschiedliche Softwarekomponenten auftreten.

Applikationen für Endanwender

z.B. Mailclients (Outlook, Thunderbird, ...)
LDAP in der Regel vor Benutzer „versteckt“

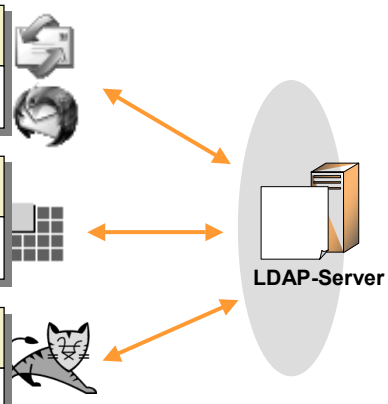
Anwendungen für Spezialisten / Administratoren

Herstellerspezifische oder –unabhängige Werkzeuge, LDAP ist unmittelbar „erlebbar“

Serverlösungen, die Verzeichnisse integrieren

z.B. Mailserver, Webserver, ...
LDAP-Kenntnisse zur Konfiguration erforderlich

...



Aufgrund einiger Besonderheiten kommen moderne Verzeichnisprodukte in vielen Unternehmen zum Einsatz.

Einige Besonderheiten von Verzeichnissen

- Optimierung auf Suchoperationen und Lesezugriffe
- Möglichkeit der automatischen Bildung von Repliken, dadurch z.B. bessere Zugriffszeiten in geographisch verteilten Organisationen
- Verweise („Referrals“, wörtlich = Empfehlungen) ermöglichen verteilte Verzeichnisse auf standardisierte Weise
- Standardisierung des Informationsmodells und vorgefertigter „Schemata“, sowie des TCP/IP-basierten Zugriffs (LDAP)

Anwendungsbeispiele

- Zentrale Verwaltung von Ressourcen im Netzwerk (Drucker, Arbeitsplatzrechner, Dienste, ...)
- Zentrale Verwaltung von Benutzerdaten, inkl. Organisationsstruktur und Berechtigungen (z.B. auf obige Ressourcen)
- Verwendung der Daten für Bestandslisten, Telefonbücher (online/offline), Generierung von Organigrammen

LDAP. Eine Einführung für Java-Entwickler
oose. Innovative Informatik

Einige LDAP-Serverprodukte

Microsoft
Sun microsystems
IBM
OpenLDAP®
fedora directory server

Kommerzielle Server (Auswahl):

- Microsoft Active Directory
- Novell eDirectory
- Sun Java System Directory Server
- IBM Tivoli Directory Server
- …

Open Source (Auswahl):

- OpenLDAP
- Fedora Directory Server
- Apache Directory Server
- …

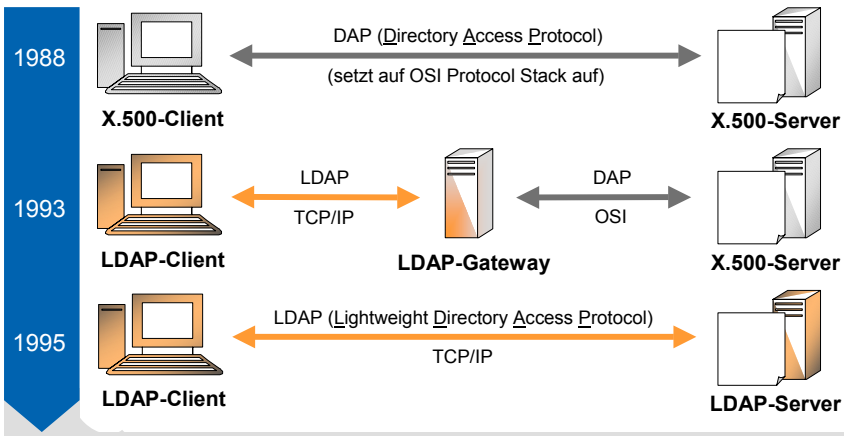
2

➔ Was ist LDAP überhaupt? Ein paar (!) Details.

- Geschichte
- Informationsmodell, Operationen
- Suchen in Verzeichnissen

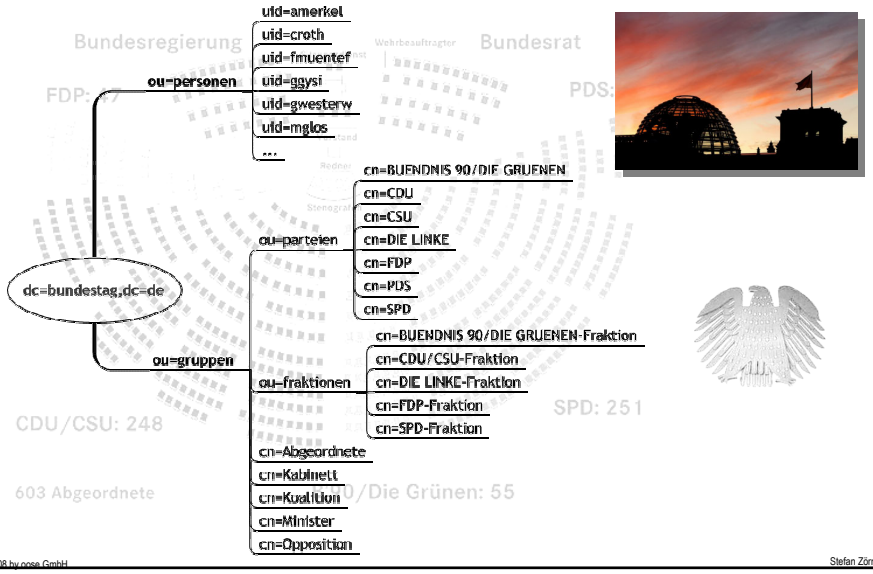
LDAP. Eine Einführung für Java-Entwickler
oose. Innovative Informatik

In der historischen Entwicklung war LDAP zunächst nur als IP-basierte Zugriffsoption für X.500-Verzeichnisse gedacht.



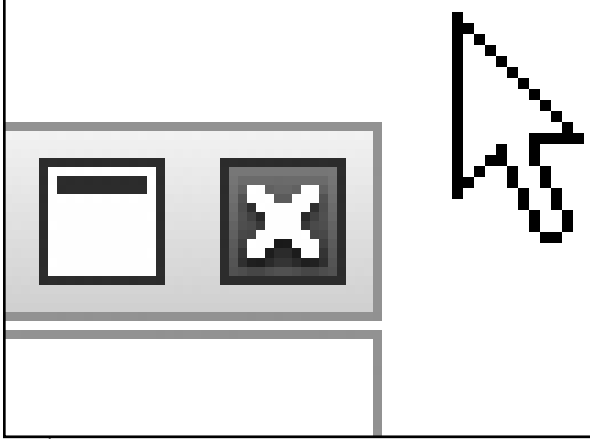
1988: X.500 Standard für Verzeichnisse, 1993: LDAPv1, 1995: LDAPv2,
1995: erster nativer LDAP-Server (University of Michigan), 1996: Netscape Directory Server
1996: LDAPv3 (RFC 2251-2256), aktuelle Fassung von 2006 (RFC 4510)

Als Beispielinhalte betrachten wir Bundestagsabgeordnete mit Gruppenzugehörigkeiten (z.B. Kabinett, Parteien ...)



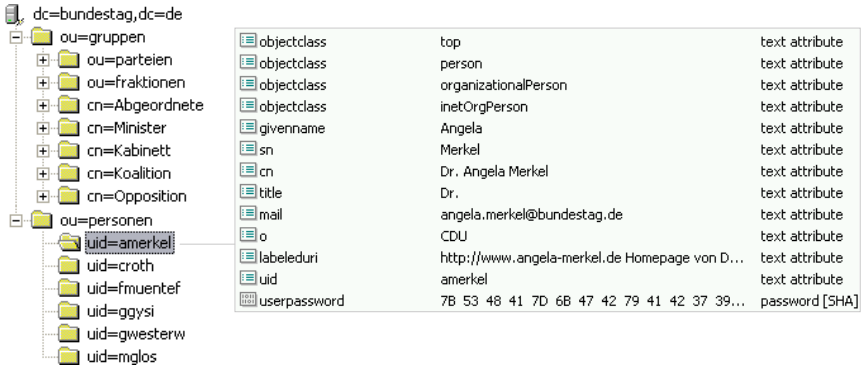
LDAP. Eine Einführung für Java-Entwickler
oose. Innovative Informatik

Demo: Stöbern in einem Verzeichnis mit LDAP Tools



Stefan Zörner – LDAP

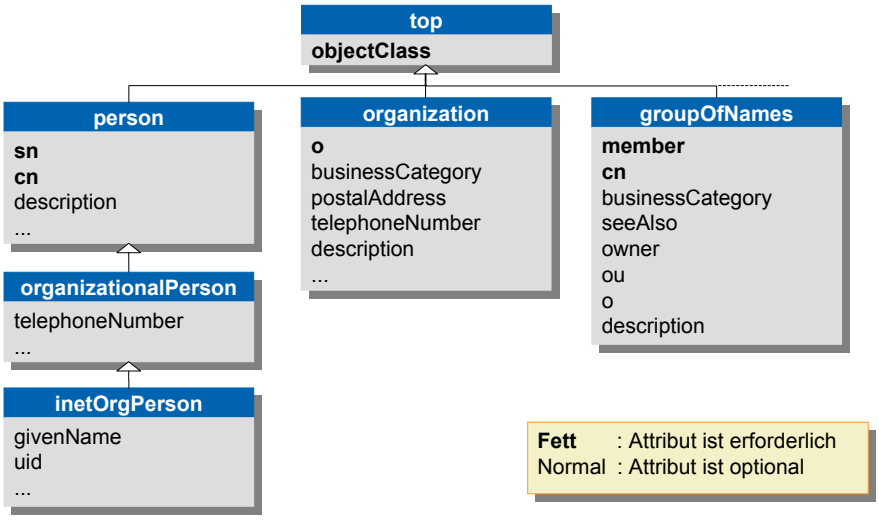
Attribute eines Eintrags im Beispielverzeichnis. So genannte RDNs bilden den eindeutigen Namen im Verzeichnis (DN).



(Relative) Distinguished Name (R(DN))

- Ein Attributwert im Eintrag ist besonders ausgezeichnet: er legt den eindeutigen Namen auf der Ebene des Baums fest (RDN)
- Die Kette der RDN von einem Eintrag bis zur Wurzel bildet den innerhalb eines Verzeichnisses eindeutigen Namen (DN) des Eintrags.
- Hier: DN=*„uid=amerkel,ou=personen,dc=bundestag,dc=de“*

Ein Ausschnitt der in RFC 2256 und 2798 festgelegten Objektklassen und Attribute als eine Art Klassendiagramm.



Alle 10 Client-Operationen des LDAP v3 Protokolls

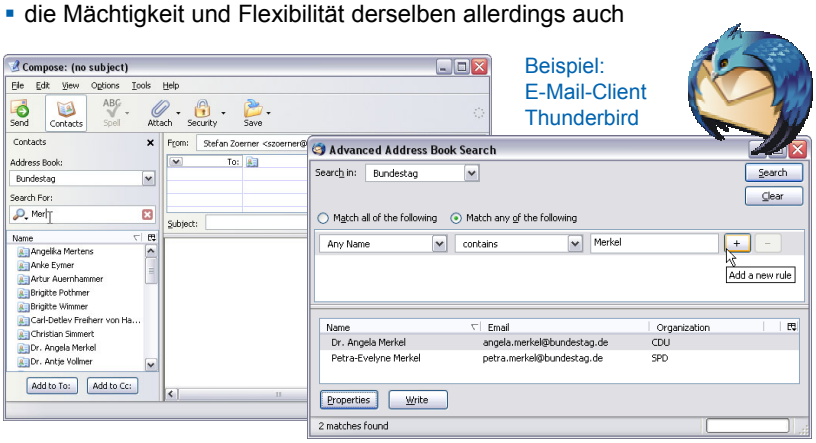
Ähnlich wie SQL kennt LDAP Operationen zum Anlegen, Ändern und Löschen von Einträgen, sowie zum Suchen.

Name	Funktion
Bind	Übermittlung von Authentifizierungsinformationen an den Server, Beginn einer Sitzung
Unbind	Beenden einer Sitzung
Search	Suchen im Verzeichnis
Add	Hinzufügen eines neuen Eintrages
Delete	Löschen eines bestehenden Eintrages
Modify	Ändern von Attributen eines bestehenden Eintrages
Modify DN	Umbenennen eines bestehenden Eintrages, Verschieben innerhalb des Verzeichnisses
Compare	Test eines Attributwertes eines bestimmten Eintrages
Abandon	Abbrechen einer zuvor abgesetzten Operation
Extended	Aufruf einer serverspezifischen Operation, die nicht im Standard beschrieben ist

LDAP. Eine Einführung für Java-Entwickler
oose. Innovative Informatik

Suche in LDAP-Verzeichnissen mit Endbenutzersoftware

- Oberfläche (Formulare) zugeschnitten auf konkrete Aufgabe
- Formulierung von Suchkriterien ist möglichst einfach gestaltet
- LDAP-spezifische Syntax und Parameter bleiben dem Benutzer verborgen
- die Mächtigkeit und Flexibilität derselben allerdings auch



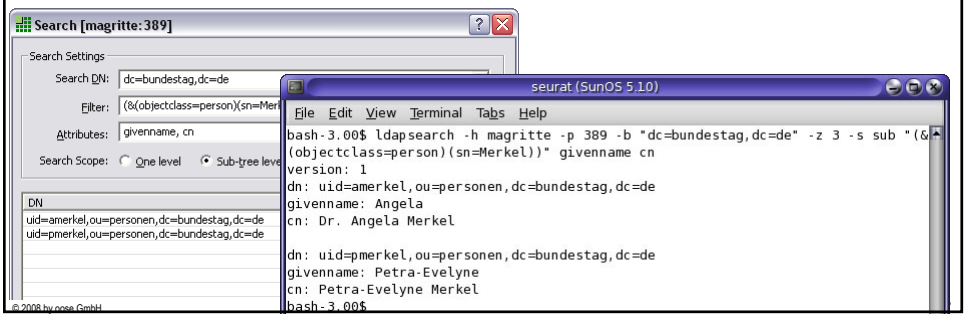
Beispiel:
E-Mail-Client
Thunderbird

© 2008 by oose GmbH
Stefan Zörner – LDAP

Suchen mit LDAP-Syntax und -Parametern

Wann werden Suchoperationen in LDAP-Syntax abgesetzt?

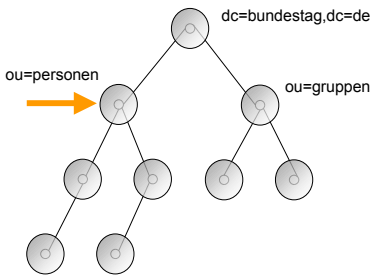
- Kommandozeilentools (gängiger Befehl: ldapsearch)
- LDAP-Clientanwendungen, Tools zur Administration (z.B. Softerra LDAP Browser/Administrator, Apache Directory Studio …)
- Bei Individualentwicklung, d.h. Verwendung einer entsprechenden API
- Konfigurationen für Produkte, die LDAP-Verzeichnisse integrieren, z.B. als Benutzerdatenbasis von Applikationsservern



Bestimmte Angaben werden bei einer Suche spezifiziert, um den Umfang der betrachteten Menge einzuschränken.

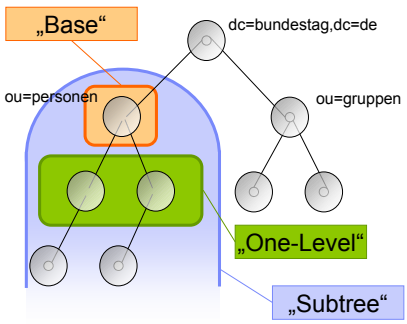
Search Base

- Eintrag, bei dem die Suche gestartet wird
- Ergebnisse liegen niemals oberhalb dieses Eintrages



Search Scope

- Knotenmenge, die bei der Suche betrachtet wird
- Bei „Base“ lediglich ein Eintrag



LDAP kennt verschiedene auf Attribute anwendbare Filterarten, um die Ergebnismenge einzugrenzen.

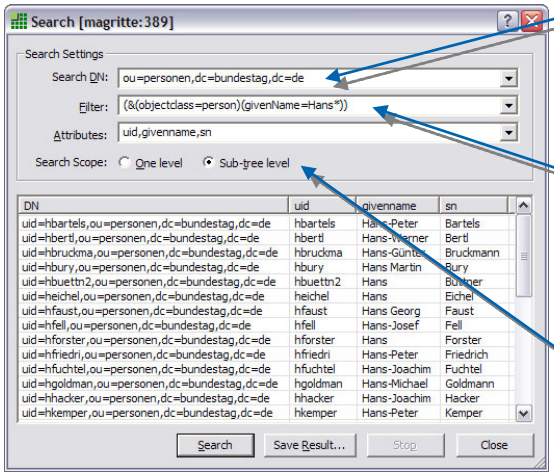
Filter	Operator	Beispiel	Bedeutung
Vorhandensein	=*	*(mail=*)*	Passt auf alle Einträge, wo das Attribut mindestens einmal vorliegt
Gleichheit	=	*(sn=Brandt)*	Passt auf alle Einträge, wo eines der Attributaufkommen exakt diesen Wert hat
Teilstrings	=	*(sn=W*)*	Passt auf alle Einträge, die auf das Muster passen (kein vollwertiges Patternmatching!)
Ordnungen	>=, <=	*(sn>=M)*	Passt auf alle der Ordnungsrelation entsprechenden Einträge
Ähnlichkeit	~=	*(sn~=Brant)*	Anwendung eines serverspezifischen Ähnlichkeitsalgorithmus (z.B. Soundex)

Kombination mit booleschen Verknüpfungen in Prefix-Schreibweise

- & für UND-Verknüpfung, d.h. *(& (Filter 1) (Filter 2) ... (Filter n))*
- | für ODER-Verknüpfung, d.h. *(| (Filter 1) (Filter 2) ... (Filter n))*
- ! für Negierung, d.h. *(! (Filter))*

LDAP. Eine Einführung für Java-Entwickler
oose. Innovative Informatik

In diesem Beispiel wird Softerra LDAP-Browser verwendet, um Suchkriterien zu spezifizieren, und zu suchen.



Search Base

Der konkrete Eintrag „ou=personen,o=bundestag.de “ im Verzeichnis

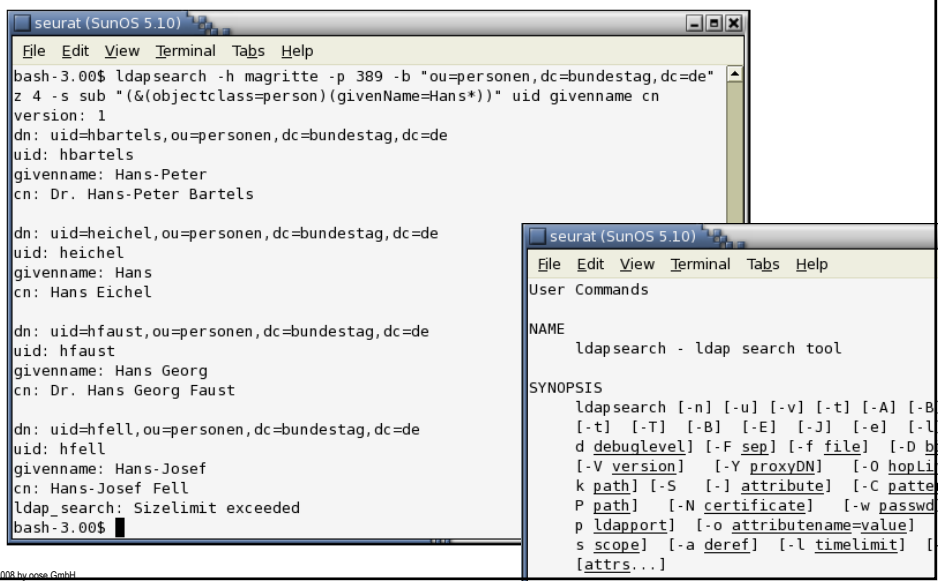
Search Filter

(&(objectclass=person) (givenname=Hans*)) Einträge, die Personen repräsentieren, deren Vorname mit Hans beginnt

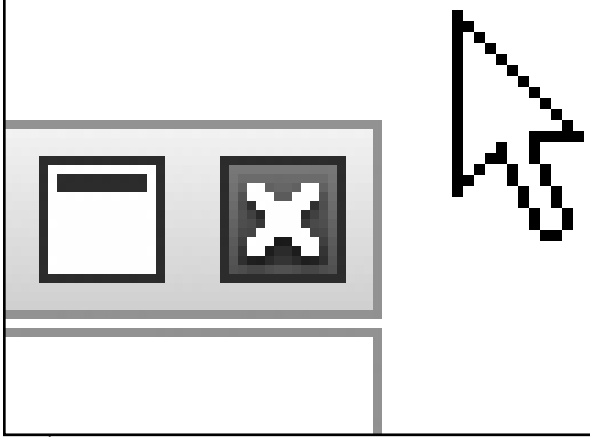
Search Scope

„sub-tree“, d.h. der gesamte Teilbaum unterhalb des Eintrags der Search Base (inklusive)

Die gleichen Parameter werden auch von gängigen Kommandozeilentools erwartet (hier: ldapsearch).



Demo: Suchen in einem Verzeichnis mit LDAP Tools



➔ Wie spricht man zu LDAP mit Java?

- Optionen im Überblick
- Native Bibliotheken
- Java Naming and Directory Interface

Aus einem Java-Programm heraus gibt es verschiedene APIs bzw. Optionen für einen LDAP-Zugriff.

Verwendung expliziter LDAP-Bibliotheken

- Implementierung von LDAP-Funktionalität unmittelbar auf Basis der Netzwerkfähigkeiten von Java (TCP/IP, Sockets, *java.net*-Package)
- Ergebnis sind APIs, welche den LDAP-Konzepten in Klassen/Schnittstellen und Methodennamen sehr nahe kommen

JNDI (Java Naming and Directory Interface)

- Programmierschnittstelle (API) von Sun zum einheitlichen Zugriff auf verschiedenste Namens- und Verzeichnisdienste, u.a. LDAP-Server
- Abstraktion von LDAP-Konzepten

DSMLv2.0 (Directory Services Markup Language)

- XML-Dokumente beschreiben Operationen auf dem Verzeichnis und die Resultate (Suchergebnisse, Fehlermeldungen, etc.)
- Kommunikation erfolgt nicht über LDAP, sondern z.B. eingebettet in SOAP über HTTP oder Message oriented Middleware

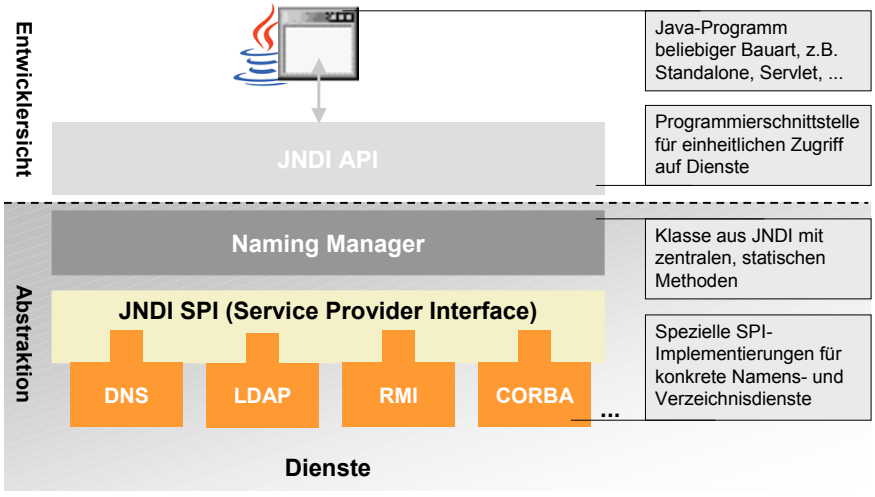
Explizite LDAP-Bibliotheken werden z.B. von Netscape und Novell angeboten, sind aber universell verwendbar.

Produkt	Directory SDK for Java	LDAP Classes for Java
Ursprung	Netscape Inc.	Novell Inc.
Bezug jetzt	www.mozilla.org/directory/	www.openldap.org/jldap/
Setzt voraus	JRE >= 1.1.7	JRE >= 1.2
unterstützt	LDAP v2, v3	LDAP v3 (v2)

Mögliche Motivationen zur Verwendung

- Kenntnisse in LDAP oder sogar in klassischen LDAP-APIs vorhanden (z.B. C, Perl), daher geringere Einarbeitungszeit als beim abstrakteren JNDI
- Zugriff auf fortgeschrittene, spezielle Funktionalitäten weitaus direkter möglich (z.B. Schemaoperationen), teilweise sogar ausschließlich (z.B. LDIF)

Die JNDI-Architektur sieht einen Plugin-Mechanismus vor – verschiedene Implementierungen bei gleich bleibender API.



Die zur JNDI API zugehörigen Komponenten sind in Packages unterhalb von *javax.naming* zu finden.

Die Darstellung beinhaltet die wesentlichen Pakete.

javax.naming

Klassen und Schnittstellen zum Zugriff auf Namensdienste

javax.naming.directory

Erweiterung des Zugriff auf Verzeichnisse

javax.naming.ldap

Spezifische Schnittstellen für das LDAP-V3-Protokoll

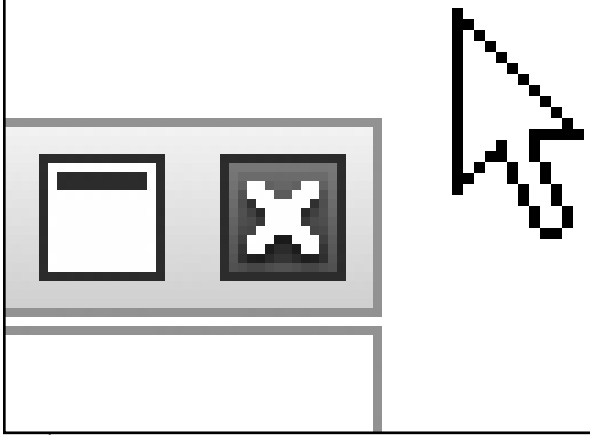
javax.naming.spi

Schnittstelle zur Realisierung von Service Providern

Bei der Konfiguration mit Properties stellen Schlüssel/Wert-Paare die Eigenschaften zur SPI bereit.

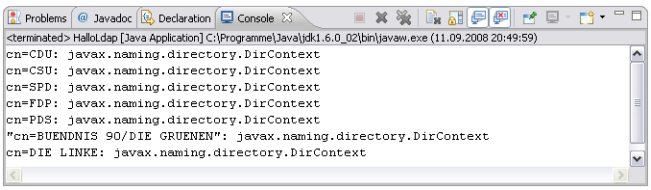
```
import java.util.Properties;
import javax.naming.Context;
import javax.naming.InitialContext;
import javax.naming.NameClassPair;
import javax.naming.NamingEnumeration;
import javax.naming.NamingException;

public class HalloLdap {
  public static void main(String[] args) throws NamingException {

    Properties env = new Properties();
    env.put(Context.INITIAL_CONTEXT_FACTORY,
      "com.sun.jndi.ldap.LdapCtxFactory");
    env.put(Context.PROVIDER_URL, "ldap://magritte:389/dc=bundestag,dc=de");
    InitialContext ctx = new InitialContext(env);

    NamingEnumeration<NameClassPair> iter =
      ctx.list("ou=parteien,ou=gruppen");
    while (iter.hasMore()) {
      System.out.println(iter.next());
    }
  }
}
```

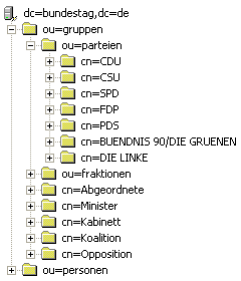
Demo: Zugriff auf ein Verzeichnis mit JNDI



Das vorherige Beispiel führt eine „anonyme“ Anmeldung durch – weitere Angaben zur Authentifizierung sind möglich.

Ausgabe des Beispiels:





- Nicht jeder LDAP-Server ist so konfiguriert, dass er anonyme Verbindungen zulässt – und selbst wenn werden nur lesende Operationen möglich sein
- Erweiterung um Angaben für Authentifizierung mit User/Passwort

```
env.put(Context.SECURITY_AUTHENTICATION, "simple");
env.put(Context.SECURITY_PRINCIPAL,
      "uid=amerkel,ou=personen,dc=bundestag,dc=de");
env.put(Context.SECURITY_CREDENTIALS, "Kanzlerin123");
```

Alternative zu dynamischen Properties: jndi.properties

Ressource-Datei *jndi.properties*:

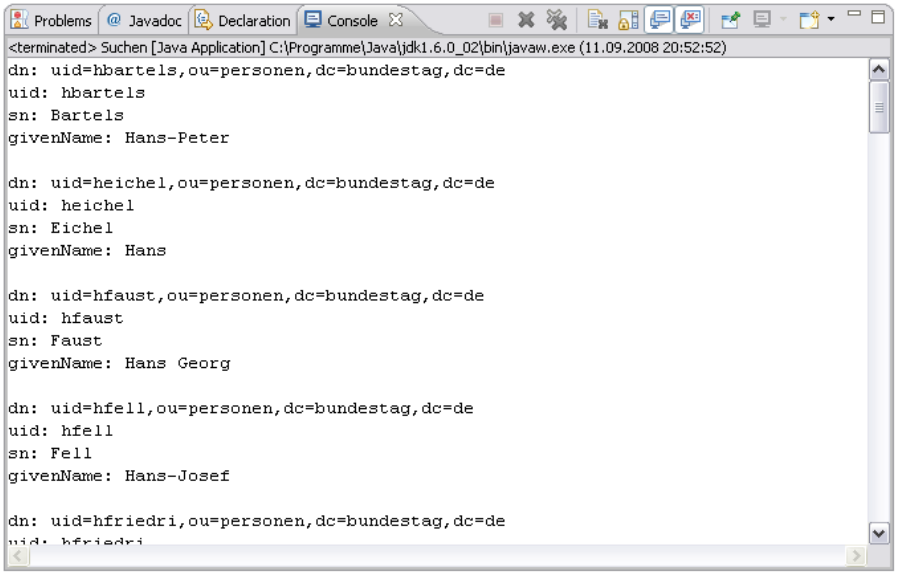
- Informationen zur Konfiguration von JNDI analog zur Hashtable/Properties
- Datei *jndi.properties* muss sich im Classpath befinden; wird dann automatisch beim Erzeugen des `InitialContext` herangezogen
- Schlüssel: Zeichenketten, die den Werten der Konstanten aus der Schnittstelle `Context` entsprechen (siehe javadoc)

Beispielinhalt der Datei

```
java.naming.factory.initial=com.sun.jndi.ldap.LdapCtxFactory
java.naming.provider.url=ldap://magritte:389/dc=bundestag,dc=de
java.naming.security.authentication=simple
java.naming.security.principal=uid=amerkel,ou=personen,dc=bundestag,dc=de
java.naming.security.credentials=Kanzlerin123
```


JNDI-Beispiel für eine Suchoperation

```
import javax.naming.*;
import javax.naming.directory.*;
...
DirContext ctx = new InitialDirContext(env);

SearchControls ctls = new SearchControls();
ctls.setSearchScope(SearchControls.SUBTREE_SCOPE);
ctls.setReturningAttributes(new String[] {"uid", "givenName", "sn"});

NamingEnumeration<SearchResult> enm = ctx.search("ou=personen",
        "(&(objectClass=person)(givenName=Hans*))", ctls);
while (enm.hasMore()) {
  SearchResult sr = enm.next();
  System.out.println("dn: " + sr.getNameInNamespace());
  Attributes attr = sr.getAttributes();
  System.out.println(attr.get("uid"));
  System.out.println(attr.get("sn"));
  System.out.println(attr.get("givenName"));
  System.out.println();
}

...
```



➔ Wie integriert man LDAP als Benutzerdatenbasis?

Security in Java EE

Konfiguration am Beispiel Tomcat

Ablauf

Bei Webapplikationen gemäß Java EE werden Security-Anforderungen deklariert, die Laufzeitumgebung konfiguriert.

Generelles Muster der Java EE

- Basisdienste (z.B. Transaktionen, Persistenz) werden durch Container bereitgestellt – Gilt auch für Security

➔ Entwicklung der Komponenten unabhängig von konkreter Laufzeitumgebung

Deklarative Security bei Webapplikationen (*web.xml*)

- Innerhalb des Deployment Descriptor können Anforderungen zu folgenden Themen deklariert werden (unabhängig von der Laufzeitumgebung):
  - Login-Konfiguration (BASIC, formbasiert, Client-Zertifikate, ...)
  - Schützenswerte Ressourcen (bzgl. Autorisierung und Transport)
- Zur Inbetriebnahme muss die Laufzeitumgebung konfiguriert werden (z.B. SSL für Vertraulichkeit, Benutzerdatenbasis für Authentifizierung).

Im Folgenden zeigen wir Konfigurationsbeispiele zur LDAP-Integration exemplarisch für Apache Tomcat 6.0.

Realms in Apache Tomcat

- Sog. Realms schlagen die Brücke zwischen Konfiguration in *web.xml* und konkretem Speicher mit Benutzerinformationen
- Tomcat unterstützt verschiedene Realms (z.B. JAAS, JDBC); auch die Implementierung eigener Realms ist möglich
- Vereinbart werden können Realms an verschiedenen Stellen der Konfigurationsdatei *server.xml*
- Details: „Tomcat Realm Configuration HOW-TO“

➔ http://tomcat.apache.org/tomcat-6.0-doc/realm-howto.html

JNDI-Realm für LDAP-Integration

- Implementierung, die JNDI‘s LDAP-Provider nutzt
- Vielfältige Konfigurationsmöglichkeiten über Attributangaben

Die im Folgenden gezeigte LDAP-Integration ist auf andere Softwareprodukte (Applikationsserver u.a.) übertragbar.

Um ein Verzeichnis zu integrieren, müssen bestimmte Informationen bereitstehen / Entscheidungen gefällt werden.

☑ **a) Verbindungsdaten zum LDAP-Server**

- Hostname, Port, ggf. Base DN
  z.B. ldap://magritte:389/dc=bundestag,dc=de
- Anonym vs. konkreter Benutzer
- Verwendung von SSL/TLS (ja/nein), falls ja: Serverzertifikat

☑ **b) Identifizierung und Authentifizierung der Benutzer**

- Wie wird von den Benutzerangaben (z.B. UID/Kennwort) auf den zugehörigen Eintrag im Verzeichnis geschlossen?
- Mit welchem Verfahren wird der Benutzer authentifiziert?

☑ **c) Zuordnung der Rollen**

- Wie kann ermittelt werden, welche Rollen ein Benutzer hat
- Wenn Einträge (z.B. Gruppen) diese Rollen repräsentieren, wie erfolgt die Zuordnung der (Java EE-) Rollennamen

Wir gehen im Folgenden davon aus, dass Benutzerkennung und Kennwort als Parameter vorliegen.

Optionen zur Identifizierung des Benutzereintrages

(1) Angabe eines Musters für den DN mit Platzhalter für Benutzerkennung

z.B. uid={0},ou=personen,dc=bundestag,dc=de

Angabe „mglos“ führt zu uid=mglos,ou=personen,dc=bundestag,dc=de

(2) LDAP-Search mit parametrisiertem Filter (flexibler)

Z.B. Search Base = „dc=bundestag,dc=de“, Scope = „Subtree“, Filter:

(&(objectclass=person)(uid={0}))

Optionen zur Authentifizierung des Benutzers

(1) Bind am LDAP-Server mit gefundenem DN und gegebenen Kennwort (falls die LDAP-Operation fehlschlägt, wird der Benutzer abgelehnt)

(2) Vergleich des angegebenen Kennwortes mit dem entsprechenden Attribut des gefundenen Benutzereintrages

Für die Identifizierung der Rollen ist entscheidend, wie die entsprechenden Informationen im Verzeichnis vorliegen.

Optionen zur Speicherung von Rollen im Verzeichnis

(1) Rollen sind als spezielle Attributwerte am Benutzereintrag gespeichert
➔ Es sind lediglich die Attributwerte auszulesen

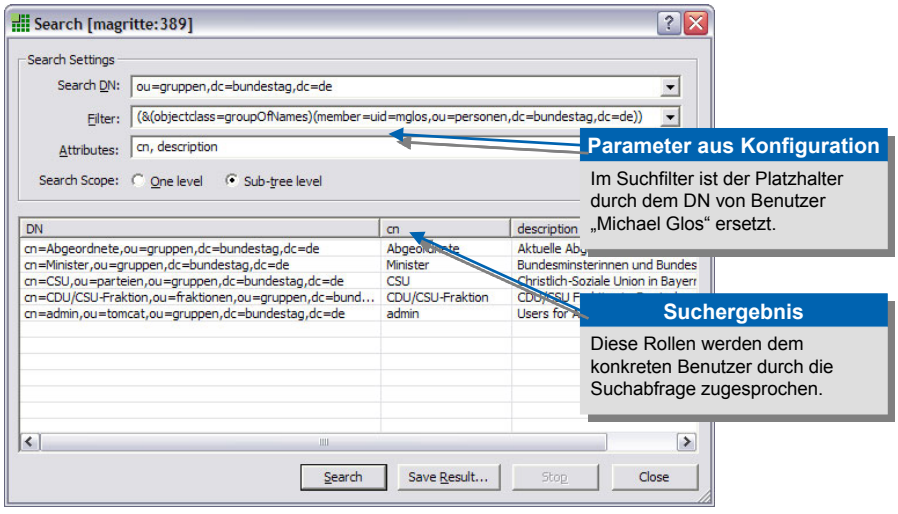
(2) Rollen entsprechen Einträgen, welche die Mitglieder aufzählen
➔ Eine geeignete Suchoperation liefert die Rollen

Eine entsprechende Suchoperation im Beispielverzeichnis

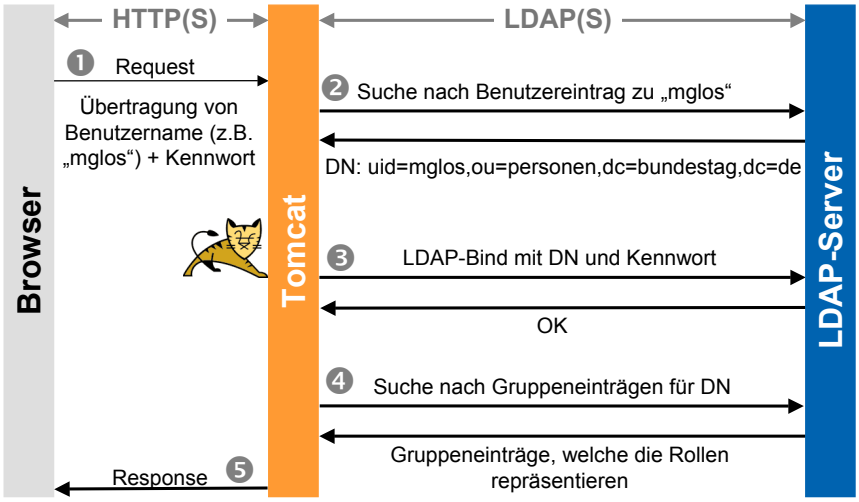
- Search Base ou=gruppen,dc=bundestag,dc=de
- Search Scope: Subtree
- Filter, in den Platzhalter wird der DN des Benutzers eingesetzt
  (&(objectclass=groupOfNames)(member={0}))

➔ Liefert alle Gruppeneinträge, die den Benutzer direkt als Mitglied enthalten. Das Attribut cn kann als Rollennamen für das Mapping in Java EE dienen.

Die Suchabfragen lassen sich in einem LDAP-Tool entwickeln und überprüfen. Hier am Beispiel der Rollen:



Zusammenfassend läuft zwischen einem Browser, Tomcat und dem LDAP-Server dann folgende Kommunikation ab.

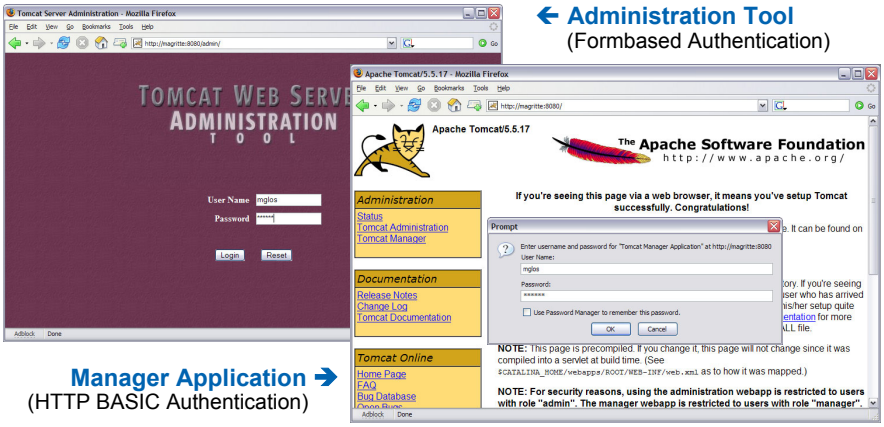


In der Konfigurationsdatei „server.xml“ von Tomcat stellen sich entsprechende Einstellungen wie folgt dar:

```
...
<Realm

      className      = "org.apache.catalina.realm.JNDIRealm"
a)    connectionURL  = "ldap://magritte:389"
      contextFactory = "com.sun.jndi.ldap.LdapCtxFactory"

      userBase       = "ou=personen,dc=bundestag,dc=de"
b)    userSubtree    = "true"
      userSearch     = "(&amp;(objectclass=person)(uid={0}))"

      roleBase       = "ou=gruppen,dc=bundestag,dc=de"
      roleSubtree    = "true"
c)    roleSearch     = "(&amp;(objectclass=groupOfNames)(member={0}))"
      roleName       = "cn"

/>

...
```

In Suchfiltern „&“ durch „&“ ersetzen.

Nach erfolgreicher Konfiguration greift Tomcat für Authentifizierung und Autorisierung auf das Verzeichnis zu.



← **Administration Tool**
(Formbased Authentication)

Manager Application →
(HTTP BASIC Authentication)

An der Konfiguration können alle deployten Anwendungen teilhaben, nicht nur diese beiden Beispiele.

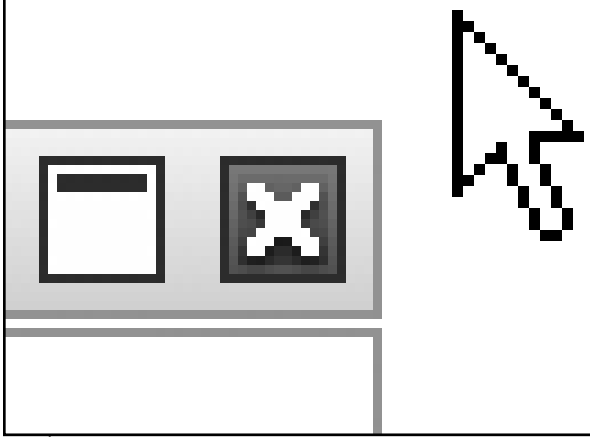
LOG-File-Auschnitt ...

```
2008-08-31  08:45:24 JNDIRealm[Catalina]: Connecting to URL ldaps://magritte:636/
2008-08-31  08:46:02 JNDIRealm[Catalina]:   entry found for oschily with dn uid=oschily,ou=personen,dc=bundestag,dc=de
2008-08-31  08:46:02 JNDIRealm[Catalina]:   validating credentials by binding as the user
2008-08-31  08:46:02 JNDIRealm[Catalina]:   binding as uid=oschily,ou=personen,dc=bundestag,dc=de
2008-08-31  08:46:02 JNDIRealm[Catalina]: Username oschily successfully authenticated
2008-08-31  08:46:02 JNDIRealm[Catalina]:   getRoles(uid=oschily,ou=personen,dc=bundestag,dc=de)
2008-08-31  08:46:02 JNDIRealm[Catalina]:   Searching role base 'ou=gruppen,dc=bundestag,dc=de' for attribute 'cn'
2008-08-31  08:46:02 JNDIRealm[Catalina]:   With filter expression
                        '(&(objectclass=groupOfNames)(member=uid=oschily,ou=personen,dc=bundestag,dc=de))'
2008-08-31  08:46:02 JNDIRealm[Catalina]:   retrieving values for attribute cn
2008-08-31  08:46:02 JNDIRealm[Catalina]:   retrieving values for attribute cn
2008-08-31  08:46:02 JNDIRealm[Catalina]:   retrieving values for attribute cn
2008-08-31  08:46:02 JNDIRealm[Catalina]:   retrieving values for attribute cn
2008-08-31  08:46:02 JNDIRealm[Catalina]:   retrieving values for attribute cn
2008-08-31  08:46:02 JNDIRealm[Catalina]:   Returning 5 roles
2008-08-31  08:46:02 JNDIRealm[Catalina]:   Found role Abgeordnete
2008-08-31  08:46:02 JNDIRealm[Catalina]:   Found role Kabinett
2008-08-31  08:46:02 JNDIRealm[Catalina]:   Found role SPD
2008-08-31  08:46:02 JNDIRealm[Catalina]:   Found role SPD-Fraktion
2008-08-31  08:46:02 JNDIRealm[Catalina]:   Found role admin
2008-08-31  08:46:23 JNDIRealm[Catalina]:   entry found for ppau with dn uid=ppau,ou=personen,dc=bundestag,dc=de
2008-08-31  08:46:23 JNDIRealm[Catalina]:   validating credentials by binding as the user
2008-08-31  08:46:23 JNDIRealm[Catalina]:   binding as uid=ppau,ou=personen,dc=bundestag,dc=de
2008-08-31  08:46:23 JNDIRealm[Catalina]:   bind attempt failed
2008-08-31  08:46:23 JNDIRealm[Catalina]:   entry found for ppau with dn uid=ppau,ou=personen,dc=bundestag,dc=de
2008-08-31  08:46:23 JNDIRealm[Catalina]:   validating credentials by binding as the user
2008-08-31  08:46:23 JNDIRealm[Catalina]:   binding as uid=ppau,ou=personen,dc=bundestag,dc=de
2008-08-31  08:46:23 JNDIRealm[Catalina]: Username ppau successfully authenticated
2008-08-31  08:46:23 JNDIRealm[Catalina]:   getRoles(uid=ppau,ou=personen,dc=bundestag,dc=de)
2008-08-31  08:46:23 JNDIRealm[Catalina]:   Searching role base 'ou=gruppen,dc=bundestag,dc=de' for attribute 'cn'
2008-08-31  08:46:23 JNDIRealm[Catalina]:   With filter expression
                        '(&(objectclass=groupOfNames)(member=uid=ppau,ou=personen,dc=bundestag,dc=de))'
2008-08-31  08:46:23 JNDIRealm[Catalina]:   retrieving values for attribute cn
2008-08-31  08:46:23 JNDIRealm[Catalina]:   retrieving values for attribute cn
2008-08-31  08:46:23 JNDIRealm[Catalina]:   retrieving values for attribute cn
2008-08-31  08:46:23 JNDIRealm[Catalina]:   Returning 3 roles
2008-08-31  08:46:23 JNDIRealm[Catalina]:   Found role Abgeordnete
2008-08-31  08:46:23 JNDIRealm[Catalina]:   Found role PDS
2008-08-31  08:46:23 JNDIRealm[Catalina]:   Found role Opposition
```

Erfolgreiche Anmeldung als „oschily“.

Anmeldung als „ppau“, 1x mit falschem, 1x mit richtigem Kennwort

Demo: LDAP-Verzeichnis als Benutzerdatenbasis für Tomcat



➔ Wenn Sie neugierig geworden sind ...

- Einige LDAP Server
- Apache Directory Studio
- Literatur

Kommerzielle LDAP-Server (Auswahl).

Zum Sammeln praktischer Erfahrungen ist die Arbeit mit einem konkreten Server-Produkt unumgänglich.

Sun Java System Directory Server:

- http://wwws.sun.com/software/

  Vormals iPlanet DS/Sun ONE DS, basiert auf Netscape

Microsoft Active Directory:

- http://www.microsoft.com/ad/

  Integraler Bestandteil der Windows 2000+ Architektur

IBM Tivoli Directory Server:

- http://www.ibm.com/software/tivoli/products/directory-server/

  Setzt auf DB2 als Datenspeicher auf

Weitere Anbieter:

Novell („eDirectory"), Oracle („Internet Directory"), Red Hat ...

Open Source LDAP-Server (Auswahl).

Neben dem Klassiker OpenLDAP sind in den letzten Jahren weitere freie Alternativen entstanden, bzw. im Entstehen begriffen.

OpenLDAP:

- http://www.openldap.org/

  Basiert auf dem LDAP-Server der University of Michigan

Fedora Directory Server:

- http://directory.fedora.redhat.com/

  Basiert auf dem Netscape Directory Server

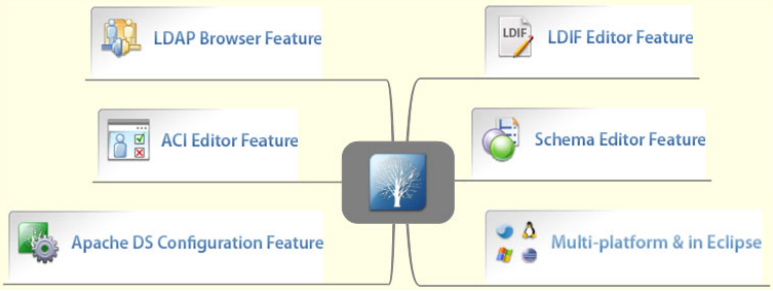
Apache Directory Server:

- http://directory.apache.org/

  100% Pure Java, einbettbar in andere Java-Komponenten

Apache Directory Studio

- Eclipse-basierter LDAP-Client
- Arbeitet mit allen gängigen Servern zusammen
- LDAP-Browser/Editor, Schema-Editor, LDIF-Editor …
- Läuft standalone (RCP) und als Plugin in einer IDE



Directory Studio: Ein paar Zahlen …

- Erster Release Februar 2007
- Erster Major Release: September 2007
- Eclipse Community Award 2008 Finalist
  (Best Open Source RCP Application)
- Aktuell Version 1.2.0: August 2008
- Seit 2007 mehr als 100.000 Downloads (!)

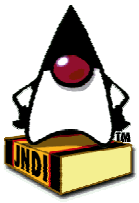
➔ http://directory.apache.org/studio/downloads.html

JNDI Tutorial – Die herausragende Online-Quelle zum Java Naming and Directory Interface ...

The JNDI Tutorial

➔ http://java.sun.com/products/jndi/tutorial/

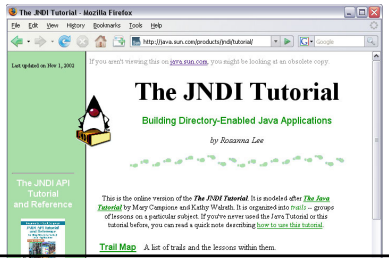
Umfangreich, aber recht alt.



Trail: Java Naming and Directory Interface

➔ http://java.sun.com/docs/books/tutorial/jndi/

Verkürzte Fassung, aber mit Neuerungen zu Java 5 und 6

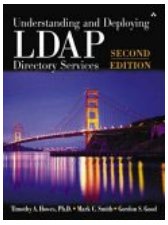


Für Interessierte bietet der Buchmarkt zahlreiche Angebote, um die Kenntnisse im Bereich LDAP zu vertiefen.

LDAP für Java-Entwickler Einstieg und Integration

Stefan Zörner
252 Seiten; Entwickler.Press;
3. aktualisierte Auflage (November 2007)
ISBN 978-3-939084-07-5

➔ http://www.entwickler-press.de/



Understanding and Deploying LDAP Directory Services

von Timothy A. Howes, Mark C. Smith, Gordon S. Good
936 Seiten
Addison-Wesley Professional, Mai 2003 (2. Auflage)
ISBN 0-672323-16-8

➔ http://awprofessional.com/title/0672323168

oose.
Innovative Informatik

Vielen Dank!

Ich freue mich auf Ihre Fragen …

Stefan Zörner :: Stefan.Zoerner@oose.de