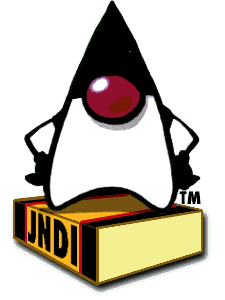
|  |  |
| --- | --- |
|  | Documentation |
|  | Class: 5AHITT  Last modification: 11/25/2014  Document Version: 1.0 |

Dezentrale Systeme

2014/2015

Christian Bobek, Osman Özsoy

Java Naming and Directory Interface



List of contents

[Statement of task 1](#_Toc404261678)

[Apportionment of work with effort estimation 2](#_Toc404261679)

[Final time apportionment 3](#_Toc404261680)

[Design consideration 4](#_Toc404261681)

[Technology description 5](#_Toc404261682)

[Task execution 6](#_Toc404261683)

[Setting up a new naming service 6](#_Toc404261684)

[Bibliography 9](#_Toc404261685)

# Statement of task

Follow the introduction and instructions of the **Java Naming and Directory Interface (JNDI)** described in this tutorial <https://docs.oracle.com/javase/tutorial/jndi/TOC.html>.  
  
Setup your own or use an pre-existing naming service and implement following operations:

* Lookup an Object (1)
* List the Context (1)
* Add, Replace or Remove a Binding (2)
* Rename (1)
* Create and Destroy Subcontexts (1)
* Attribute Names (1)
* Read Attributes (1)
* Modify Attributes (1)
* Add, Replace Bindings with Attributes (2)
* Search  
  \* Basic Search (1)  
  \* Filters (1)  
  \* Scope (1)  
  \* Result Count (1)  
  \* Time Limit (1)

Create a protocol in which you describe 1) your **installation steps**, 2)**source code snippets** used to perform the naming operations and 3)**result of each operation**. Pack the protocol and sources into a **JAR file** and upload it here.

Size of Group: **2 persons**  
(if you work without a team member then I will recommend to use the name service of another group)

# Apportionment of work with effort estimation

|  |  |  |  |
| --- | --- | --- | --- |
| Competent  person(s) | Task | Description | Estimated time in h |
| Bobek, Özsoy | Setting up a naming service or preparing a pre-existing naming service | The description of each task is specified in the given tutorial: <https://docs.oracle.com/javase/tutorial/jndi/TOC.html> | 2 |
| Özsoy | Lookup an Object | 1 |
| Bobek | List the Context | 1 |
| Özsoy | Add, Replace or Remove a Binding | 1 |
| Bobek | Rename | 1 |
| Özsoy | Create and Destroy Subcontexts | 1 |
| Bobek | Attribute Names | 1 |
| Özsoy | Read Attributes | 1 |
| Bobek | Modify Attributes | 1 |
| Özsoy | Add, Replace Bindings with Attributes | 1 |
| Bobek | Basic Search | 1 |
| Özsoy | Filters | 1 |
| Bobek | Scope | 1 |
| Özsoy | Result Count | 1 |
| Bobek | Time Limit | 1 |
| Bobek, Özsoy | Creating a protocol | Create a protocol in which you describe 1) your **installation steps**, 2) **source code snippets** used to perform the naming operations and 3) **result of each operation**. | 2 |

**Estimated total time exposure**

|  |  |
| --- | --- |
| Person | Time exposure in h |
| Bobek | 11 |
| Özsoy | 11 |
| **Sum:** | **22** |

# Final time apportionment

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Competent  person(s) | Task | Estimated time in h | Actual time in h | Comment |
| Bobek, Özsoy | Setting up a naming service or preparing a pre-existing naming service | 2 | 6 | The setup of a new naming service was very difficult, so it takes more time to get it successfully work |
| Özsoy | Lookup an Object | 1 |  |  |
| Bobek | List the Context | 1 |  |  |
| Özsoy | Add, Replace or Remove a Binding | 1 |  |  |
| Bobek | Rename | 1 |  |  |
| Özsoy | Create and Destroy Subcontexts | 1 |  |  |
| Bobek | Attribute Names | 1 |  |  |
| Özsoy | Read Attributes | 1 |  |  |
| Bobek | Modify Attributes | 1 |  |  |
| Özsoy | Add, Replace Bindings with Attributes | 1 |  |  |
| Bobek | Basic Search | 1 |  |  |
| Özsoy | Filters | 1 |  |  |
| Bobek | Scope | 1 |  |  |
| Özsoy | Result Count | 1 |  |  |
| Bobek | Time Limit | 1 |  |  |
| Bobek, Özsoy | Creating a protocol | 2 |  |  |

**Actual total time exposure**

|  |  |
| --- | --- |
| Person | Time exposure in h |
| Bobek |  |
| Özsoy |  |
| **Sum:** |  |

# Design consideration

# Technology description

# Task execution

## Setting up a new naming service

The first step in our task execution was **setting up a new naming service**. We decided to do the setup in a virtual machine, which we imported in VMware Workstation. The operating system of the virtual machine was **debian**.

We had to install and configure **OpenLDAP**, which is our naming service. The following installation and configuration steps were necessary to get OpenLDAP work:

Updating the VM:

# apt-get update

Installing the OpenLDAP server:

# apt-get install slapd ldap-utils libldap-2.4-2 libdb4.6

During the installation process, we are requested to determine a password for the **admin entry**. So we set the admin password to: **admin**

Once the installation-process is finished, then the OpenLDAP server should run successfully.

Standardly OpenLDAP uses the database named “olcDatabase={1}mdb.ldif”, which is located in “/etc/ldap/slapd.d/cn=config”.

To manage the entries of the used database on the OpenLDAP server, we have to install **JXplorer** under debian:

# apt-get install jexplorer

To start JXplorer, use the following command:

# jxplorer

The properties below are standardly determined (by the installation-process):

dc=nodomain

cn=admin

We changed the domain name (dc) to “**jndi\_dezsys**”. To do this, we had to change all of the “**dc=nodomain**” properties in the file “/etc/ldap/slapd.d/cn=config/olcDatabase={1}mdb.ldif” to “**dc=jndi\_dezsys**”.

**Consider:** Before modifying this file, stop OpenLDAP server with following command:

# service slapd stop

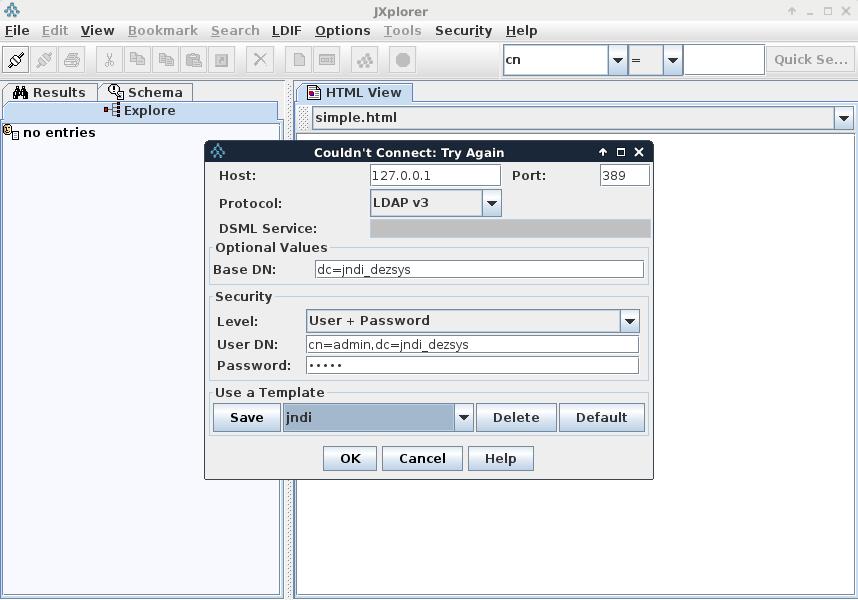
After finishing the modification start the server again with:

# service slapd start

After we have configured all necessary properties, we started **JXplorer to manage our entries** in our OpenLDAP database. To connect to our database over JXplorer, we have to push the **connect-button** and type in mandatory connection information.

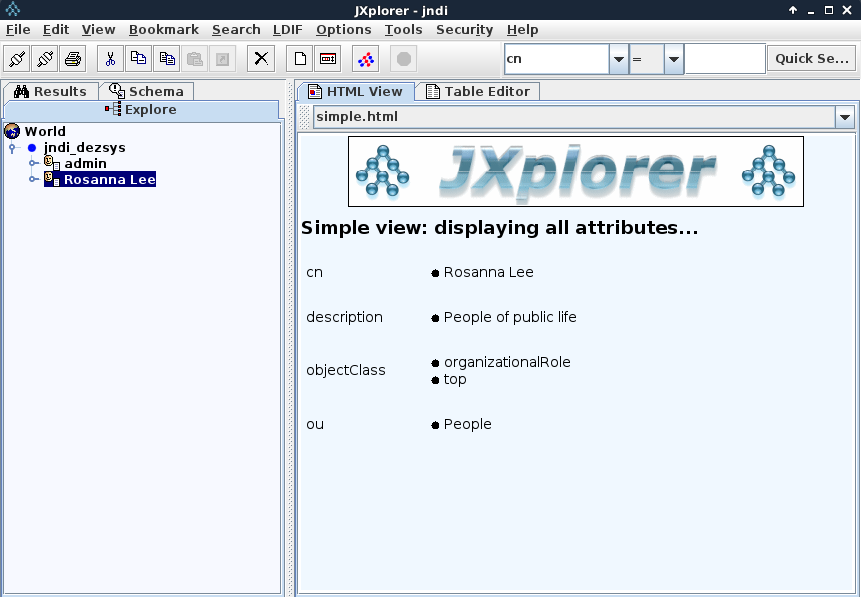
Once you have filled in the information, you can save it as a **template** and use this saved template when reconnecting to the same database.

Our **connection-template** in JXplorer looks as below obvious (we named the template “**jndi**”):



After successful connection to the database, we should be able to **add**, **delete** or **modify** **entries** of the **database**. The “**admin**”-entry should be **standardly** in the database.

In the next step, we created a new entry named “**Rosanna Lee**” by right-clicking “**jndi\_dezsys**” and selecting the **“New entry”-option**:



# Bibliography

|  |  |
| --- | --- |
| [1] | Title: Trail: Java Naming and Directory Interface: Table of Contents  Author: Oracle  Online-/Resource: https://docs.oracle.com/javase/tutorial/jndi/TOC.html  last modified: /  abstracted: 11/25/2014 |
| [2] | Title: New York University Computer Science Department Courant Institute of Mathematical Sciences  Author: Jean-Claude Franchitti  Online-/Resource: http://www.nyu.edu/classes/jcf/g22.3033-007\_sp01/handouts/g22\_3033\_h83.htm  last modified:  abstracted: |
| [3] | Title: OpenLDAP installation on Debian  Author: docelic  Online-/Resource: https://www.debian-administration.org/article/585/OpenLDAP\_installation\_on\_Debian  last modified: 03/19/2008  abstracted: 11/19/2014 |
| [4] | Title: Database Creation and Maintenance Tools  Author: OpenLDAP  Online-/Resource: http://www.openldap.org/doc/admin24/dbtools.html  last modified: /  abstracted: 11/19/2014 |
| [5] | Title:  Author:  Online-/Resource:  last modified:  abstracted: |