

Solution of Exercise Sheet 8

Exercise 1 (Web Services)

1. What is a web service?
(Explain in just a few sentences.)

A *Web service* is a software system, identified by a *Uniform Resource Identifier (URI)*, whose public interfaces and bindings are defined and described using *XML*. Its definition can be discovered by other software systems. These systems may then interact with the *Web service* in a manner prescribed by its definition, using *XML* based messages conveyed by *Internet protocols*.

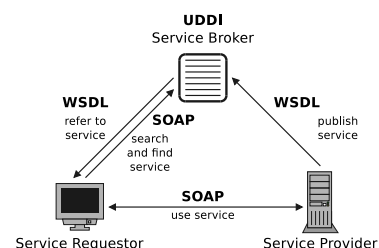
2. What is XML-RPC?

It is a simple and language independent protocol for implementing remote procedure calls via *HTTP*. *XML* is used to specify the input parameters for the remote procedures in a well-defined structure. Servers and clients can be implemented in any programming language. Function calls are transmitted via *HTTP POST* to the server. The server evaluates the *XML* document, which is contained in the message, and uses its content as parameter for calling the desired function. The result is written into a *XML* document and transmitted to the client.

3. Which three roles contains the theoretical implementation of SOAP web services?

The theoretical implementation includes these three roles:

- *Service provider*
- *Service requestor (user / customer)*
- *Service registry (broker)*



4. Which markup language is used by SOAP web services for interaction?

XML

5. What is WSDL and for what purpose is it used?

WSDL (Web Services Description Language) is an *XML*-based interface description language to specify...

- *how a web service can be called*
- *what parameters a web service expects*
- *what data structures a web service returns*

6. What is UDDI and for what purpose is it used?

UDDI (Universal Description, Discovery and Integration) provides a standardized directory structure for the metadata (e.g. technical characteristics, requirements and provider information) of web services.

7. Explain the difference between UDDI and WS-Inspection.

UDDI = few, centralized directories, where different providers publish their services.

WS-Inspection = many decentralized, small directories, in which few providers publish their services.

8. Describe the difference between the theoretical implementation of SOAP web services and the way, SOAP web services operate in practice.

Publicly accessible web services are usually offered without using UDDI. The reason is that individual web services and the required access information are already known to the user or client application. For this reason, the users and client applications no longer need to search for a web services in a registry service.

9. Which protocol is used by RESTful web services for interaction?

HTTP

10. Which four methods are sufficient to initiate all necessary functions on objects inside RESTful web services?

(Name the methods and explain the purpose of each method.)

- *PUT/POST requests to create or replace a resource.*
- *GET requests to download a resource.*
- *PUT requests to modify a resource.*
- *DELETE requests to erase a resource.*

11. Which two methods are often used addition to the methods of subtask 10?

(Name the methods and explain the purpose of each method.)

- *HEAD requests metadata about a resource (file) from the server. This way, metadata of a resource can be requested, without transferring the resource itself. The same header is returned as with GET.*
- *OPTIONS requests the information, which methods are supported by a resource.*

Exercise 2 (RESTful Web Services)

1. Do some basic interaction with the Python library boto [1] and an cloud infrastructure service of your choice.
2. Check your requests and the replies of the infrastructure service via Wireshark [2] to understand the way of interaction.

[1] <https://github.com/boto/boto>

[2] <http://www.wireshark.org/>

This simple python script helps you with your first steps. It fetches the list of buckets of a user from Amazon S3.

```
1 #!/usr/bin/env python
2 from boto.ec2.connection import *
3 from boto.s3.connection import *
4
5 try:
6     calling_format=boto.s3.connection.OrdinaryCallingFormat()
7     connection = boto.s3.connection.S3Connection(
8         aws_access_key_id="<ACCESS_KEY>",
9         aws_secret_access_key="<SECRET_ACCESS_KEY>",
10        is_secure=False,
11        validate_certs=False,
12        host="s3.amazonaws.com",
13        calling_format=calling_format,
14        path="/")
15 except S3ResponseError:
16     print "Error!"
17 else:
18     print "Connection established."
19
20 requestbuckets = connection.get_all_buckets()
21 print (requestbuckets)
22
23 for entry in requestbuckets:
24     print entry.name
25     print entry.creation_date
26     print entry.get_acl
27     # If the bucket is located anywhere other than inside region us-east-1,
28     # the location is printed next to it's name. Otherwise, it will be blank
29     print entry.get_location()
```

Exercise 3 (Private Cloud Storage Service)

Install a Private Cloud storage service, which implements the S3 API. You can use e.g. one of these solutions:

- OpenStack Swift [3]
- Eucalyptus Walrus [4]
- Nimbus Cumulus [5]
- S3 ninja [6]

You can also use any other solution, which is free software. You can install the storage service natively or inside a virtual machine.

- [3] <http://www.openstack.org>
- [4] <http://www.eucalyptus.com>
- [5] <http://www.nimbusproject.org>
- [6] <http://www.s3ninja.net>

Installation of a Nimbus Cumulus S3 storage service in Debian 8.5

These sources are helpful for understanding the installation.

- <http://www.nimbusproject.org/docs/current/admin/z2c/service-setup.html>
- And the file `/nimbus/cumulus/docs/QUICKSTART.txt` inside the source code repository: <http://www.nimbusproject.org/downloads/nimbus-iaas-2.10.1-src.tar.gz>

The file `QUICKSTART.txt` describes the installation and how to use the Cumulus service alone (without the ability to deploy virtual machines).

1. Install Packages:

```
1 $ sudo apt-get install -y curl, htop, joe, nmap, virtualenv,  
2 python-dateutil, python-pysqlite2, sqlite3, git, build-essential,  
3 openjdk-6-jdk, libssl-dev, ant-optional, curl, python-dev
```

2. Create user nimbus:

```
1 $ sudo adduser nimbus
```

3. Switch to user nimbus and relocate to its home directory:

```
1 $ su nimbus  
2 $ cd ~
```

4. Fetch the nimbus sources and unpack them:

```
1 $ curl -O http://www.nimbusproject.org/downloads/nimbus-iaas-2.10.1-  
   src.tar.gz  
2 $ tar xvzf nimbus-iaas-2.10.1-src.tar.gz
```

5. Create a new directory for cumulus and install the S3 service there:

```
1 $ mkdir ~/cumulus  
2 $ ./nimbus-iaas-2.10.1-src/cumulus/cumulus-install.sh ~/cumulus
```

6. Attach this line into the file `~/.bashrc`

```
1 export CUMULUS_HOME=~/cumulus
```

7. Start a new bash:

```
1 $ bash
```

The config file of cumulus is here: ~/cumulus/etc/cumulus.ini

8. Start cumulus:

```
1 $ ~/cumulus/bin/cumulus &
```

Now, the new service should be available at port 8888:

```
1 $ nmap localhost
2
3 Starting Nmap 6.47 ( http://nmap.org ) at 2016-06-19 14:30 EDT
4 Nmap scan report for localhost (127.0.0.1)
5 Host is up (0.00034s latency).
6 Other addresses for localhost (not scanned): 127.0.0.1
7 Not shown: 996 closed ports
8 PORT      STATE SERVICE
9 22/tcp    open  ssh
10 25/tcp    open  smtp
11 111/tcp   open  rpcbind
12 8888/tcp   open  sun-answerbook
```

9. Create a new user:

```
1 $ ~/cumulus/bin/cumulus-add-user christianbaun@gmail.com
2 ID           : YrziRoQJHaLSc2bGMLLVf
3 password     : eqKi2rYZyObyMwBxaEOZdc5QZphiBbw01M1CoMTv82
4 quota       : None
5 canonical id : 33f2bcde-3643-11e6-baa7-000c29f809f2
```

Now, the S3-compatible storage service is ready to use, but next, we need some tools do work with it:

10. Option 1: s3cmd (<http://s3tools.org/s3cmd>)

Execute as user nimbus:

```
1 $ cd ~
2 $ wget http://downloads.sourceforge.net/project/s3tools/s3cmd/1.6.1/
   s3cmd-1.6.1.tar.gz
3 $ tar xvzf s3cmd-1.6.1.tar.gz
4 $ cp ~/cumulus/etc/dot_s3cfg.in ~/.s3cfg
```

Insert inside the file ~/.s3cfg inside these lines...

```
1 access_key = @@ID@@
2 host_base = @@HOST_PORT@@
3 host_bucket = @@HOST_PORT@@
4 secret_key = @@KEY@@
5 use_https = @@SEC@@
```

the correct values:

```
1 access_key = YrziRoQJHaLSc2bGMMLVf
2 host_base = localhost:8888
3 host_bucket = localhost:8888
4 secret_key = eqKi2rYZy0byMwBxaEOZdc5QZphiBbw01M1CoMTv82
5 use_https = False
```

Insert into the file /home/nimbus/.bashrc these lines:

```
1 export PATH=${PATH}:~/s3cmd-1.6.1
```

And start a new bash:

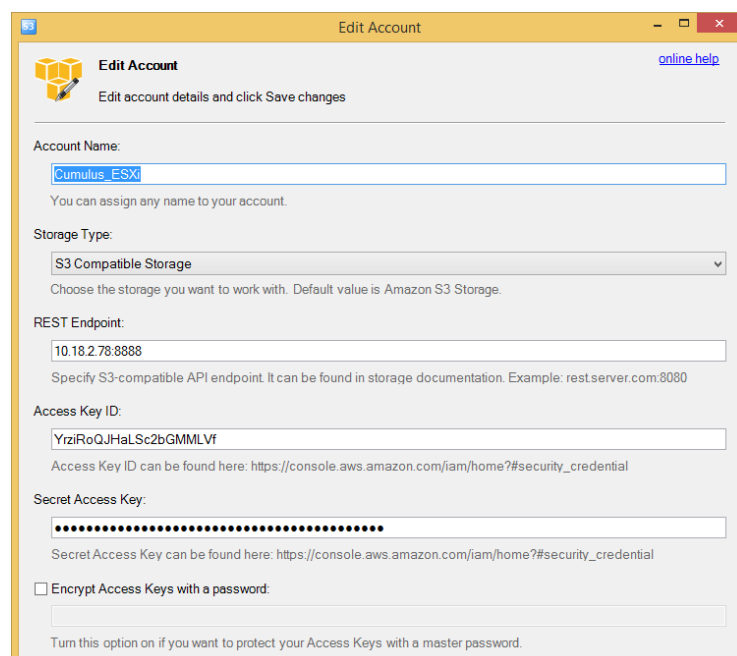
```
1 $ bash
```

Now, you can work with the service via the command s3cmd.

```
1 $ s3cmd mb s3://TEST
2 $ s3cmd put /etc/group s3://TEST/group
3 $ s3cmd ls s3://TEST/
```

11. Option 2: S3 Browser (<http://s3browser.com>)

This is a windows tool. Using it is quite simple. For using it, it is mandatory to insert the correct credentials:



Exercise 4 (Web Service Interaction)

Develop an application with the boto library, which interacts with the Private Cloud storage service, you have already installed. The application should...

- fetch and print out the list of your buckets
- create a bucket
- upload a local file into the new bucket
- fetch and print out the list of objects in the new bucket
- download the previously uploaded object from the bucket
- erase the previously uploaded object in the new bucket
- erase the new bucket

To simplify the task, you can hard code the file name and bucket name.

TBD