

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ
НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ФАКУЛЬТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

Отчет по лабораторной работе №4
по курсу «Распределенные системы»

Выполнил: студент 1-го курса магистратуры, гр. 21224

Гафиятуллин А.Р

Новосибирск, 2022

1. ОПРЕДЕЛЕНИЯ

2. СОКРАЩЕНИЯ

DCMES — Dublin Core Metatdata Element Set

API — Application Programming Interface

3. ОБЪЕКТ ИССЛЕДОВАНИЯ

Программный интерфейс **ZooPARK-DS API**.

4. ЦЕЛИ РАБОТЫ

Создать программу-клиент для работы с сервером **ZooPARK-DS** на основе программного интерфейса **ZooPARK-DS API**.

5. МЕТОДОЛОГИЯ/ПОРЯДОК ПРОВЕДЕНИЯ РАБОТ

5.1. Развернута виртуальная машина.

5.2. Установлен программный пакет **zoopark-ds-1.0.3**:

```
libuser@ds104:~/tmp/zoopark-ds-1.0.3 % make install
Making install in src
../config/install-sh -c -d '/home/libuser/local/bin'
/bin/sh ../libtool --mode=install /usr/bin/install -c zebra-client zebra-worker zebrash '/home/libuser/local/bin'
libtool: install: /usr/bin/install -c zebra-client /home/libuser/local/bin/zebra-client
libtool: install: /usr/bin/install -c zebra-worker /home/libuser/local/bin/zebra-worker
libtool: install: /usr/bin/install -c zebrash /home/libuser/local/bin/zebrash
Making install in include
Making install in cuba
../config/install-sh -c -d '/home/libuser/local/include/zoopark-ds'
/usr/bin/install -c -m 644 ctrl.h funupdate.h param.h version.h worker.h '/home/libuser/local/include/zoopark-ds'
Making install in etc
../config/install-sh -c -d '/home/libuser/local/etc/zoopark-ds'
/usr/bin/install -c -m 644 rc.conf.dist zebra-worker-start.sh zebra-worker-stop.sh yazgfs.xml cql2pqf.txt '/home/libuser/local/etc/zoopark-ds'
Making install in profiles
Making install in dom
Making install in tab
../config/install-sh -c -d '/home/libuser/local/share/zoopark-ds/profiles/dom/tab'
/usr/bin/install -c -m 644 string.chr urx.chr dom.xml extract.xsl input.xsl store2dc.xsl '/home/libuser/local/share/zoopark-ds/profiles/dom/tab'
../config/install-sh -c -d '/home/libuser/local/share/zoopark-ds/profiles/dom'
/usr/bin/install -c -m 644 zebra.cfg zebra.json '/home/libuser/local/share/zoopark-ds/profiles/dom'
Making install in fit.nsu.ru
Making install in tab
../config/install-sh -c -d '/home/libuser/local/share/zoopark-ds/profiles/fit.nsu.ru/tab'
/usr/bin/install -c -m 644 string.chr urx.chr dom.xml extract.xsl input.xsl '/home/libuser/local/share/zoopark-ds/profiles/fit.nsu.ru/tab'
../config/install-sh -c -d '/home/libuser/local/share/zoopark-ds/profiles/fit.nsu.ru'
/usr/bin/install -c -m 644 zebra.cfg zebra.json '/home/libuser/local/share/zoopark-ds/profiles/fit.nsu.ru'
Making install in tests
Making install in dom
Making install in fit.nsu.ru
libuser@ds104:~/tmp/zoopark-ds-1.0.3 %
```

5.3. Установлен программный пакет **zoopark-ds-api-1.0.0**:

```
libuser@ds104:~/repos/zoopark-ds-api % yarn install
yarn install v1.22.17
[1/4] Resolving packages...
[2/4] Fetching packages...
[3/4] Linking dependencies...
warning " > bookshelf@1.2.0" has incorrect peer dependency "knex@>=0.15.0 <0.22.0".
[4/4] Building fresh packages...
warning Your current version of Yarn is out of date. The latest version is "1.22.18", while you're on "1.22.17".
Done in 98.70s.
libuser@ds104:~/repos/zoopark-ds-api %
```

5.4. Тестовая коллекция записей **DCMES** в XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<dc:collection xmlns="http://fit.nsu.ru/ds" xmlns:dc="http://purl.org/dc/elements/1.1/">
  <dc:metadata>
    <dc:identifier>https://www.nsu.ru</dc:identifier>
    <dc:title>Новосибирский государственный университет</dc:title>
    <dc:title>НГУ</dc:title>
  </dc:metadata>
  <dc:metadata>
    <dc:identifier>https://www.nsu.ru/n/information-technologies-department/</dc:identifier>
    <dc:title>
      Факультет информационных технологий Новосибирского государственного университета
    </dc:title>
    <dc:title>ФИТ НГУ</dc:title>
  </dc:metadata>
</dc:collection>
```

5.5. Профиль базы данных **fit.nsu.ru** для записей DCMES в XML:

```
libuser@ds104:~/tmp/zoopark-ds-1.0.3/profiles/fit.nsu.ru % ls -lR
total 76
-rw-r--r--  1 libuser  libuser   5296  2 апр.  02:05 cql2pqf.txt
-rw-r--r--  1 libuser  libuser  22428 19 апр.  23:35 Makefile
-rw-r--r--  1 libuser  libuser   187  2 апр.  02:04 Makefile.am
-rw-r--r--  1 libuser  libuser  21234  2 апр.  18:25 Makefile.in
drwxr-xr-x  2 libuser  libuser    512 19 апр.  23:35 tab
-rw-r--r--  1 libuser  libuser    14  2 апр.  01:06 zebra.cfg
-rw-r--r--  1 libuser  libuser   979 19 апр.  23:35 zebra.json
-rw-r--r--  1 libuser  libuser   943  2 апр.  01:06 zebra.json.in

./tab:
total 56
-rw-r--r--  1 libuser  libuser    272  2 апр.  01:53 dom.xml
-rw-r--r--  1 libuser  libuser   1420 30 марта 15:36 extract.xml
-rw-r--r--  1 libuser  libuser    549 22 марта 02:32 input.xml
-rw-r--r--  1 libuser  libuser  15840 19 апр.  23:35 Makefile
-rw-r--r--  1 libuser  libuser    151  2 апр.  01:54 Makefile.am
-rw-r--r--  1 libuser  libuser  14622  2 апр.  18:25 Makefile.in
-rw-r--r--  1 libuser  libuser    842 22 марта 02:32 string.chr
-rw-r--r--  1 libuser  libuser    378 22 марта 02:32 urx.chr
```

5.6. Создана программа-клиент на основе спецификации **ZooPARK-DS API**:

Исходный код: <https://github.com/xp10rd/NSU-FIT/tree/master/masters-1st-year/distributed-systems/laboratory-work-4/client>

Клиент создан на основе клиента на Node.js и JavaScript с добавленными задачами **search** и **scan**. Исходный код задач **search** и **scan** в приложении.

5.7.Используя программу-клиент создан репозиторий и база данных на основе профиля **fit.nsu.ru**:

```
REPOS BEFORE:
{ success: true, data: [] }

CREATE REPO:
{
  success: true,
  data: {
    name: 'r1',
    type: 'fit.nsu.ru',
    id: 'e13a52f7-e89b-4615-8d8e-9aa413f24ce1',
    updated_at: '2022-05-08T13:12:03.059Z',
    created_at: '2022-05-08T13:12:03.059Z'
  }
}

CREATE DB:
{
  success: true,
  data: {
    name: 'db1',
    repository_id: 'e13a52f7-e89b-4615-8d8e-9aa413f24ce1',
    id: 'f1610df1-4929-47ba-aabe-8f45d511d5f7',
    updated_at: '2022-05-08T13:12:03.070Z',
    created_at: '2022-05-08T13:12:03.070Z'
  }
}
```

5.8.Используя программу-клиент загружена и выполнено индексирование тестовой коллекции записей:

```
CREATE STORAGE:
{
  success: true,
  data: {
    uuidfilename: 'e19c44a3-b41c-4b03-b9a4-90eb7927ad25',
    filename: 'collection.xml',
    mimetype: 'application/xml',
    filesize: 785,
    database_id: 'f1610df1-4929-47ba-aabe-8f45d511d5f7',
    addinfo: '',
    id: '0b80e703-b4cd-44c2-b473-553d70d01195',
    updated_at: '2022-05-08T13:12:03.110Z',
    created_at: '2022-05-08T13:12:03.110Z'
  }
}

UPDATE DB:
{ success: true, data: { success: true } }
```

5.9. Задача **search** с использованием языка **CQL**:

```
SEARCH RESULT:
{
  success: true,
  data: {
    success: true,
    data: {
      numberOfRecords: 2,
      records: [
        {
          recordIdentifier: null,
          recordPosition: 1,
          recordPacking: 'string',
          recordSchema: 'dc',
          recordData: '<?xml version="1.0" encoding="UTF-8"?>\n' +
            '<dc:metadata xmlns:dc="http://purl.org/dc/elements/1.1/">\n' +
            '  <dc:identifier>https://www.nsu.ru/dc:identifier>\n' +
            '  <dc:title>Новосибирский государственный университет</dc:title>\n' +
            '  <dc:title>НГУ</dc:title>\n' +
            '</dc:metadata>\n' +
            'f1610df1-4929-47ba-aabe-8f45d511d5f7'
        },
        {
          recordIdentifier: null,
          recordPosition: 2,
          recordPacking: 'string',
          recordSchema: 'dc',
          recordData: '<?xml version="1.0" encoding="UTF-8"?>\n' +
            '<dc:metadata xmlns:dc="http://purl.org/dc/elements/1.1/">\n' +
            '  <dc:identifier>https://www.nsu.ru/n/information-technologies-department/</dc:identifier>\n' +
            '  <dc:title>\n' +
            '    Факультет информационных технологий Новосибирского государственного университета\n' +
            '</dc:title>\n' +
            '  <dc:title>ФИТ НГУ</dc:title>\n' +
            '</dc:metadata>\n'
        }
      ]
    }
  }
}
```

5.10. Задача **scan** с использованием языка **CQL**:

```
SCAN RESULT:
{
  success: true,
  data: {
    success: true,
    data: {
      terms: [
        { numberOfRecords: 2, value: 'нгу', displayTerm: 'НГУ' },
        {
          numberOfRecords: 1,
          value: 'новосибирский',
          displayTerm: 'Новосибирский'
        },
        {
          numberOfRecords: 1,
          value: 'новосибирского',
          displayTerm: ''
        }
      ]
    }
  }
}
```

5.11. Используя программу-клиент удален созданный репозиторий:

```
DELETE STORAGE:  
{ success: true, data: {} }  
  
DELETE DB:  
{ success: true, data: { success: true } }  
  
DELETE REPO:  
{ success: true, data: { success: true } }  
  
REPOS AFTER:  
{ success: true, data: [] }
```

6. ОПИСАНИЕ РЕЗУЛЬТАТА

Изучен программный интерфейс **ZooPARK-DS API**, и на его основе была создана программа-клиент для создания репозитория, совершения задач **search** и **scan**, удаления репозитория для профиля базы данных для записей DCMES в XML.

7. ССЫЛКИ

1. ANSI/NISO Z39.85-2012 — URL:
<https://groups.niso.org/higherlogic/ws/public/download/10258>
2. PM2 — URL: <https://pm2.keymetrics.io/>
3. Zebra - User's Guide and Reference — URL:
<https://software.indexdata.com/zebra/doc/index.html>
4. Профиль fit.nsu.ru — URL:
https://bitbucket.org/oleg_kolobov/ds/src/master/
5. XSL Transformations (XSLT) Version 1.0 — URL:
<https://www.w3.org/TR/1999/REC-xslt-19991116>
6. XML Path Language (XPath) Version 1.0 — URL:
<https://www.w3.org/TR/1999/REC-xpath-19991116/>
7. Prefix Query Format (PQF) — URL:
<https://software.indexdata.com/yaz/doc/tools.html#PQF>
8. CQL — URL: <https://www.loc.gov/standards/sru/cql/index.html>
9. Bib-1 Attribute Set — URL:
<https://www.loc.gov/z3950/agency/defns/bib1.html>
10. Исходный код клиента — URL:
<https://github.com/xp10rd/NSU-FIT/tree/master/masters-1st-year/distributed-systems/laboratory-work-4/client>

ПРИЛОЖЕНИЕ:

1. Задача **search**:

```
const http = require('http');

const querystring = require('querystring');

exports.search = function(database_id, data, done) {
  const encodedQuery = querystring.stringify(data);

  const options = {
    hostname: process.env.HOSTNAME,
    port: process.env.PORT,
    path: process.env.APIPATH + '/databases/' + database_id + '/search?' +
encodedQuery,
    method: 'GET',
  };

  const req = http.request(options, function(res) {
    let output = null;

    res.on("data", function(chunk) {
      if (output === null) {
        output = chunk;
      } else {
        output += chunk;
      }
    });

    res.on("end", function() {
      done(null, output);
    });
  });

  req.on('error', function(error) {
    done(error, null);
  });

  req.end();
}
```

2. Использование задачи **search**:

```

.then(function(data) {
    return new Promise(function(resolve, reject) {
        requests.search(database.id, {type : "CQL", query: "dc.title= hry",
recordSchema : "dc"}, function(err, data) {
            if (err) {
                reject(err);
            } else {
                console.log("\nSEARCH RESULT:");
                console.dir(JSON.parse(data), {depth: null, colors: true});
                resolve(data);
            }
        });
    });
});
})

```

3. Задача scan:

```

exports.scan = function(database_id, data, done) {

    const encodedQuery = querystring.stringify(data);

    const options = {
        hostname: process.env.HOSTNAME,
        port: process.env.PORT,
        path: process.env.APIPATH + '/databases/' + database_id + '/scan?' +
encodedQuery,
        method: 'GET',
    };

    const req = http.request(options, function(res) {
        let output = null;

        res.on("data", function(chunk) {
            if (output === null) {
                output = chunk;
            } else {
                output += chunk;
            }
        });

        res.on("end", function() {
            done(null, output);
        });
    });

    req.on('error', function(error) {

```



```
    done(error, null);  
  });
```

```
  req.end();  
}
```

4. Использование задачи **scan**:

```
.then(function(data) {  
  return new Promise(function(resolve, reject) {  
    requests.scan(database.id, {type : "CQL", scanClause: "dc.title=нгу"},  
function(err, data) {  
  if (err) {  
    reject(err);  
  } else {  
    console.log("\nSCAN RESULT:");  
    console.dir(JSON.parse(data), {depth: null, colors: true});  
    resolve(data);  
  }  
});  
});  
})
```