МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ
НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ФАКУЛЬТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

Отчет по лабораторной работе №5

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

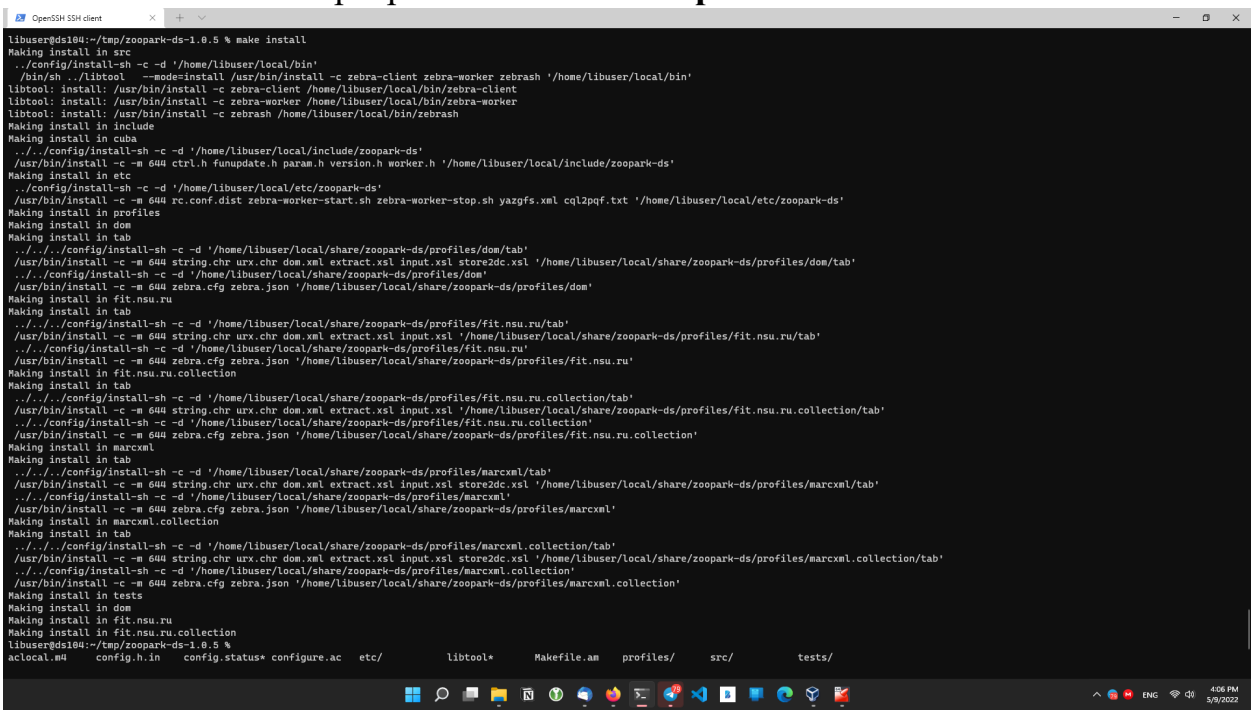
   API — Application Programming Interface
3. **ОБЪЕКТ ИССЛЕДОВАНИЯ**

   Программный интерфейс **ZooPARK-DS API**.
4. **ЦЕЛИ РАБОТЫ**

   Создать программу для синхронизации выделенной базы данных на основе других баз данных используя **ZooPARK-DS API**.
5. **МЕТОДОЛОГИЯ/ПОРЯДОК ПРОВЕДЕНИЯ РАБОТ**

   5.1. Развернута виртуальная машина.

   5.2. Установлен программный пакет **zoopark-ds-1.0.5**:



   5.3. Установлен программный пакет **zoopark-ds-api-1.0.2**:

```
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...
[1/3] . @vscode/sqlite3
[1/3] . @vscode/sqlite3
[1/3] . @vscode/sqlite3
[1/3] . @vscode/sqlite3
warning Your current version of Yarn is out of date.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 100.31s.
```

5.4. Дополнен профиль базы данных fit.nsu.ru для записей DCMES в XML элементами данных "дата добавления записи в БД" и "дата изменения записи в БД":

```
    <!-- Bib-1 1011 (Date/time-added-to-db) -->
    <xsl:template match="sys:created-at">
        <z:index name="Date/time-added-to-db:w Date/time-added-to-db:p">
            <xsl:value-of select="text()"/>
        </z:index>
    </xsl:template>
<!-- Bib-1 1012 (Date/time-last-modified) -->
    <xsl:template match="sys:updated-at">
        <z:index name="Date/time-last-modified:w Date/time-last-modified:p">
            <xsl:value-of select="text()"/>
        </z:index>
    </xsl:template>
</xsl:stylesheet>
extract.xsl: 65 строк, 1960 символов.
```

5.5. Тестовая коллекция записей **DCMES** в **XML** для базы данных **db1**:

```
<?xml version="1.0" encoding="UTF-8"?>
<dc:collection xmlns="http://fit.nsu.ru/ds"
    xmlns:dc="http://purl.org/dc/elements/1.1/"
    xmlns:sys="http://fit.nsu.ru/sys/1.0/">
    <dc:metadata>
        <dc:identifier>https://www.nsu.ru</dc:identifier>
        <dc:title>Новосибирский государственный университет</dc:title>
        <dc:title>НГУ</dc:title>
        <sys:created-at>19700101</sys:created-at>
        <sys:updated-at>20220419</sys:updated-at>
    </dc:metadata>
</dc:collection>
```

5.6. Тестовая коллекция записей **DCMES** в **XML** для базы данных **db2**:

```
<?xml version="1.0" encoding="UTF-8"?>
<dc:collection xmlns="http://fit.nsu.ru/ds"
    xmlns:dc="http://purl.org/dc/elements/1.1/"
    xmlns:sys="http://fit.nsu.ru/sys/1.0/">
    <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>
        <sys:created-at>20220420</sys:created-at>
        <sys:updated-at>20220420</sys:updated-at>
    </dc:metadata>
</dc:collection>
```

5.7. Создана программа, совмещающая оба требуемых клиента и проводящая тестирование работы.
Исходный код можно найти на https://github.com/xp10rd/NSU-FIT/blob/master/masters-1st-year/distributed-systems/laboratory-work-5/client/index.js
либо в **приложении** ниже.

5.8. Лог тестирования в **приложении**.

6. ОПИСАНИЕ РЕЗУЛЬТАТА

Создана программа для синхронизации выделенной базы данных на основе других баз данных используя ZooPARK-DS API. Проведено тестирование программы.

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/blob/master/masters-1st-year/distributed-systems/laboratory-work-5/client/index.js

ПРИЛОЖЕНИЕ:

Программа:

```
const axios = require('axios')

const fs = require('fs')
const querystring = require('querystring');
const FormData = require('form-data')

class Application {

  url = "http://localhost:3000/api/v1"

  async del_repo(id) {
    console.log(`Removed repo: ${id}`)

    const response = await axios.delete(`${this.url}/repositories/${id}`)

    if (response.status !== 200) {
      return -1
    }

    console.log(response.data)
  }

  async clear_reps() {
    console.log("Clear repos!");

    const response = await axios.get(this.url + "/repositories")

    if (response.status !== 200) {
      return -1
    }

    for (const id of response.data.data.map(r => r.id)) {
      this.del_repo(id)
    }

    return 0
  }

  async create_repo(name, type) {
    let request = { "name": name, "type": type }
```

```
  const response = await axios.post(this.url + "/repositories", request);

  if (response.status !== 201) {
    return -1
  }

  console.log(response.data)

  let res = response.data.data.id
  console.log(`Create repo ${name}: ${res}`)

  return res
}

async commit(id) {
  console.log(`Commit repo: ${id}`);

  const response = await axios.post(`${this.url}/repositories/${id}/commit`)

  if (response.status !== 200) {
    return -1
  }

  console.log(response.data)
}


async create_db(id, name) {
  let request = { "name": name, "repository_id": id }

  const response = await axios.post(this.url + "/databases", request)

  if (response.status !== 201) {
    return -1
  }

  console.log(response.data)

  let res = response.data.data.id
  console.log(`Create db ${name}: ${res}`)

  return res
}

async create_storage(databaseId, filename) {
```

```
    const formData = new FormData();
    formData.append('database_id', databaseId);
    formData.append(filename, fs.createReadStream(filename));

    const response = await axios.post(this.url + "/storages", formData, {
      headers: formData.getHeaders()
    });

    if (response.status !== 201) {
      console.log(response.data)
      return -1
    }

    console.log(response.data)
    let res = response.data.data.id
    console.log(`Create a storage for ${filename}: ${res}`);

    return res
  }

  async update_db(db_id, storage_id) {
    console.log(`Update a db ${db_id} for storage ${storage_id}`);

    const response = await axios.post(`${this.url}/databases/${db_id}/update/$
{storage_id}`)

    if (response.status !== 200) {
      console.log(response.data)
      return -1
    }

    console.log(response.data)
  }

  async search(id, query, recordSchema) {
    console.log(`Search in db: ${id}`);

    const encodedQuery = querystring.stringify({ type: "PQF", query: query,
recordSchema: recordSchema });

    const response = await axios.get(`${this.url}/databases/${id}/search?` +
encodedQuery)

    if (response.status !== 200) {
      return null
```

```
    }

    if (response.data.data.success === false) {
      return response.data.data
    }

    return response.data.data.data
  }

  async scan(id, request) {
    console.log(`Scan in db: ${id}`);

    const encodedQuery = querystring.stringify({ type: "PQF", scanClause:
request, number: "5", position: "1" });

    const response = await axios.get(`${this.url}/databases/${id}/scan?` +
encodedQuery);

    if (response.status !== 200) {
      return null
    }

    if (response.data.data.success === false) {
      return response.data.data
    }

    return response.data.data.data
  }

  async update_record(id, record) {
    const response = await axios.post(`${this.url}/databases/${id}/updateRecord`,
{ record: record })

    if (response.status !== 200) {
      console.log(response.data)
      return null
    }

    if (response.data.data.success === false) {
      return response.data.data
    }
  }

  async sync(id_1, id_2) {
    console.log(`Sync dbs: ${id_1} and ${id_2}`);
```

```
    let times = await this.scan(id_1, "@1=1011 @5=1 1970")

    for (let t of times.terms.map(t => t.displayTerm)) {
      let result = await this.search(id_1, `@1=1011 @5=1 ${t}`, "dc")

      for (let r of result.records) {
        await this.update_record(id_2, r.recordData)
      }
    }
  }

}

async function start() {
  let app = new Application();

  await app.clear_reps()

  console.log("--------------- PREPARING ---------------")

  // prepare the first db
  let repo_1 = await app.create_repo("r1", "fit.nsu.ru.collection")
  let db_1 = await app.create_db(repo_1, "db1")
  let storage_1 = await app.create_storage(db_1, "./collection-1.xml")
  await app.update_db(db_1, storage_1)

  // prepare the second db
  let db_2 = await app.create_db(repo_1, "db2")
  let storage_2 = await app.create_storage(db_2, "./collection-2.xml")
  await app.update_db(db_2, storage_2)

  await app.commit(repo_1)

  // prepare the third db
  let repo_2 = await app.create_repo("r2", "fit.nsu.ru")
  let db_3 = await app.create_db(repo_2, "db3")

  console.log("--------------- SYNC ---------------")

  // sync dbs
  await app.sync(db_1, db_3)
  await app.sync(db_2, db_3)

  console.log("--------------- TEST ---------------")
```

```
  // test
  console.log(await app.search(db_3, "@1=4 нгу", "dc"))
  console.log(await app.scan(db_3, "@1=4 нгу"))

  console.log(await app.search(db_3, "@1=4 фит", "dc"))
  console.log(await app.scan(db_3, "@1=4 фит"))

  console.log("--------------- CLEAN ---------------")

  // clean
  await app.del_repo(repo_1)
  await app.del_repo(repo_2)
}

start().then(() => console.log(''))
```

Лог тестирования:

```
Clear repos!
---------- PREPARING ----------
{
  success: true,
  data: {
    name: 'r1',
    type: 'fit.nsu.ru.collection',
    id: '22f5273e-0f70-4302-9f04-e78d64eb6000',
    updated_at: '2022-05-29T14:09:13.606Z',
    created_at: '2022-05-29T14:09:13.606Z'
  }
}
Create repo r1: 22f5273e-0f70-4302-9f04-e78d64eb6000
{
  success: true,
  data: {
    name: 'db1',
    repository_id: '22f5273e-0f70-4302-9f04-e78d64eb6000',
    id: 'b38ebfd1-bd9a-4096-8050-00b0d0483894',
    updated_at: '2022-05-29T14:09:13.619Z',
    created_at: '2022-05-29T14:09:13.619Z'
```

```
  }
}
Create db db1: b38ebfd1-bd9a-4096-8050-00b0d0483894
{
  success: true,
  data: {
    uuidfilename: '844945ea-f89b-486d-8ab6-2cffd4b14b05',
    database_id: 'b38ebfd1-bd9a-4096-8050-00b0d0483894',
    filename: 'collection-1.xml',
    mimetype: 'application/xml',
    filesize: 543,
    id: 'a16ae129-93d3-44f1-bfbc-17fa3ce03efb',
    updated_at: '2022-05-29T14:09:13.660Z',
    created_at: '2022-05-29T14:09:13.660Z',
    addinfo: null
  }
}
Create a storage for ./collection-1.xml: a16ae129-93d3-44f1-bfbc-17fa3ce03efb
Update a db b38ebfd1-bd9a-4096-8050-00b0d0483894 for storage a16ae129-
93d3-44f1-bfbc-17fa3ce03efb
{ success: true, data: { success: true } }
{
  success: true,
  data: {
    name: 'db2',
    repository_id: '22f5273e-0f70-4302-9f04-e78d64eb6000',
    id: '93ccfdfc-50ec-41e9-afed-088c006e7bf8',
    updated_at: '2022-05-29T14:09:13.688Z',
    created_at: '2022-05-29T14:09:13.688Z'
  }
}
Create db db2: 93ccfdfc-50ec-41e9-afed-088c006e7bf8
{
  success: true,
  data: {
    uuidfilename: '8e3c3952-707f-491b-9f50-5ea1f8322dc7',
    database_id: '93ccfdfc-50ec-41e9-afed-088c006e7bf8',
```

```
    filename: 'collection-2.xml',
    mimetype: 'application/xml',
    filesize: 688,
    id: '869dab36-979e-41c3-9be6-add3058917d2',
    updated_at: '2022-05-29T14:09:13.706Z',
    created_at: '2022-05-29T14:09:13.706Z',
    addinfo: null
  }
}
Create a storage for ./collection-2.xml: 869dab36-979e-41c3-9be6-add3058917d2
Update a db 93ccfdfc-50ec-41e9-afed-088c006e7bf8 for storage 869dab36-979e-
41c3-9be6-add3058917d2
{ success: true, data: { success: true } }
Commit repo: 22f5273e-0f70-4302-9f04-e78d64eb6000
{ success: true, data: { success: true } }
{
  success: true,
  data: {
    name: 'r2',
    type: 'fit.nsu.ru',
    id: '58c0dce9-0097-400d-a727-da69a58a7329',
    updated_at: '2022-05-29T14:09:13.738Z',
    created_at: '2022-05-29T14:09:13.738Z'
  }
}
Create repo r2: 58c0dce9-0097-400d-a727-da69a58a7329
{
  success: true,
  data: {
    name: 'db3',
    repository_id: '58c0dce9-0097-400d-a727-da69a58a7329',
    id: '0b3cfea4-5f5e-49b6-b392-ee83ee267c4f',
    updated_at: '2022-05-29T14:09:13.746Z',
    created_at: '2022-05-29T14:09:13.746Z'
  }
}
Create db db3: 0b3cfea4-5f5e-49b6-b392-ee83ee267c4f
```

---------- SYNC ----------
Sync dbs: b38ebfd1-bd9a-4096-8050-00b0d0483894 and 0b3cfea4-5f5e-49b6-b392-ee83ee267c4f
Scan in db: b38ebfd1-bd9a-4096-8050-00b0d0483894
Search in db: b38ebfd1-bd9a-4096-8050-00b0d0483894
Sync dbs: 93ccfdfc-50ec-41e9-afed-088c006e7bf8 and 0b3cfea4-5f5e-49b6-b392-ee83ee267c4f
Scan in db: 93ccfdfc-50ec-41e9-afed-088c006e7bf8
Search in db: 93ccfdfc-50ec-41e9-afed-088c006e7bf8
---------- TEST ----------
Search in db: 0b3cfea4-5f5e-49b6-b392-ee83ee267c4f

```
{
  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/" xmlns:sys="http://fit.nsu.ru/sys/1.0/">\n' +
        '        <dc:identifier>https://www.nsu.ru</dc:identifier>\n' +
        '        <dc:title>Новосибирский государственный университет</dc:title>\n' +
        '        <dc:title>НГУ</dc:title>\n' +
        '        <sys:created-at>19700101</sys:created-at>\n' +
        '        <sys:updated-at>20220419</sys:updated-at>\n' +
        '    </dc:metadata>\n'
    },
    {
      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/"
xmlns:sys="http://fit.nsu.ru/sys/1.0/">\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' +
      '       <sys:created-at>20220420</sys:created-at>\n' +
      '       <sys:updated-at>20220420</sys:updated-at>\n' +
      '    </dc:metadata>\n'
   }
  ]
}
Scan in db: 0b3cfea4-5f5e-49b6-b392-ee83ee267c4f
{
  terms: [
    { numberOfRecords: 2, value: 'нгу', displayTerm: 'НГУ' },
    {
      numberOfRecords: 1,
      value: 'новосибирский',
      displayTerm: 'Новосибирский'
    },
    {
      numberOfRecords: 1,
      value: 'новосибирского',
      displayTerm: 'Новосибирского'
    },
    {
      numberOfRecords: 1,
      value: 'технологий',
      displayTerm: 'технологий'
    },
    {
      numberOfRecords: 1,
      value: 'университет',
```

```
      displayTerm: 'университет'
    }
  ]
}
Search in db: 0b3cfea4-5f5e-49b6-b392-ee83ee267c4f
{
  numberOfRecords: 1,
  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/"
xmlns:sys="http://fit.nsu.ru/sys/1.0/">\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' +
        '        <sys:created-at>20220420</sys:created-at>\n' +
        '        <sys:updated-at>20220420</sys:updated-at>\n' +
        '    </dc:metadata>\n'
    }
  ]
}
Scan in db: 0b3cfea4-5f5e-49b6-b392-ee83ee267c4f
{ terms: [ { numberOfRecords: 1, value: 'фит', displayTerm: 'ФИТ' } ] }
---------- CLEAN ----------
Removed repo: 22f5273e-0f70-4302-9f04-e78d64eb6000
{ success: true, data: { success: true } }
Removed repo: 58c0dce9-0097-400d-a727-da69a58a7329
{ success: true, data: { success: true } }
```