

به نام او که از ما به شما عاشق تر است

تحلیل و طراحی چالش Data Date

گروه نرم افزاری پارت

نفیسه عامری^۱

^۱ دانشجوی ارشد هوش مصنوعی، دانشکده مهندسی، دانشگاه فردوسی، مشهد

چکیده

در این چالش قصد داریم که ابتدا با دریافت تعدادی تگ از کاربر و ارسال آن به سرور درخواست دهیم که لیستی از آبجکت‌های مورد نظر را که آن تگ‌های مدنظر کاربر را دارد، به ما برگرداند. سپس با این لیست، عملیات مرتب سازی و صفحه‌بندی را با توجه به نیاز کاربر انجام می‌دهیم. این برنامه را با زبان node.js نوشته‌ایم.

فصل ۱ – شرح تکنیکال

ابتدا با دریافت تعداد تگ از کاربر و ارسال آن به سرور درخواست می‌دهیم که لیستی از آبجکت‌های مورد نظر را که آن تگ‌های مدنظر کاربر را دارد، به ما برگرداند. به دلیل اینکه به سرور دسترسی نداریم، یک شبیه‌سازی برای سرور نوشتیم؛ به این صورت که از فایل sample.json که تعدادی آبجکت نمونه وجود دارد، می‌خوانیم و به درخواست ارسال شده پاسخ می‌دهیم.

برای پیاده‌سازی بخش اصلی برنامه، ابتدا پارامترهای Tags، SortIndex، PageSize و PageIndex از کاربر دریافت می‌کنیم. سپس متغیر Tags را به صورت جیسون به سرور ارسال کرده و خروجی آن را در DataSet ذخیره می‌کنیم.

زمانی که دریافت دیتاست از سرور به اتمام رسید، عملیات مرتب‌سازی و صفحه‌بندی را با توجه به نیاز کاربر انجام می‌دهیم.

بر اساس مقدار SortIndex که از کاربر دریافت کردیم از لیست Tagsهای DataSet بر اساس اندیس مورد نظر مرتب‌سازی را انجام می‌دهیم. بعد از مرتب‌سازی برای صفحه‌بندی آبجکت‌ها از دو حلقه تودرتو استفاده می‌کنیم و صفحه‌بندی را انجام می‌دهیم. و در انتها باتوجه به مقدار PageIndex صفحه مورد نظر کاربر را برمیگردانیم.

فصل ۲- شرح نتایج

برای تست، از تعدادی نمونه زیر استفاده کردیم.

```
[
  {
    "UUID": "fghj-98765-wdejpg-re4t5",
    "Tags": ["enc_base64", "active", "date20121218"],
    "Data": "test_2"
  },
  {
    "UUID": "fghj-98765-wdejpg-re4t5",
    "Tags": ["enc_base64", "active", "date20130618"],
    "Data": "test_4"
  },
  {
    "UUID": "fghj-98765-wdejpg-re4t5",
    "Tags": ["enc_base64", "active", "date20110128"],
    "Data": "test_3"
  },
  {
    "UUID": "fghj-98765-wdejpg-re4t5",
    "Tags": ["enc_base64", "active", "date20190823"],
    "Data": "test_4"
  },
  {
    "UUID": "fghj-98765-wdejpg-re4t5",
    "Tags": ["enc_base64", "active", "date20171017"],
    "Data": "test_3"
  },
  {
    "UUID": "fghj-98765-wdejpg-re4t5",
    "Tags": ["enc_base64", "active", "date20120815"],
    "Data": "test_1"
  },
  {
    "UUID": "fghj-98765-wdejpg-re4t5",
    "Tags": ["enc_base64", "active", "date20130327"],
    "Data": "test_4"
  },
  {
    "UUID": "fghj-98765-wdejpg-re4t5",
    "Tags": ["enc_base64", "active", "date20190923"],
    "Data": "test_4"
  },
  {
    "UUID": "fghj-98765-wdejpg-re4t5",
    "Tags": ["enc_base64", "active", "date20171117"],
    "Data": "test_3"
  },
  {
    "UUID": "fghj-98765-wdejpg-re4t5",
    "Tags": ["enc_base64", "active", "date20171017"],
    "Data": "test_3"
  },
  {
    "UUID": "fghj-98765-wdejpg-re4t5",
    "Tags": ["enc_base64", "active"],
    "Data": "test_1"
  },
  {
    "UUID": "fghj-98765-wdejpg-re4t5",
    "Tags": ["enc_base64", "active", "date20110909"],
    "Data": "test_5"
  },
  {
    "UUID": "fghj-98765-wdejpg-re4t5",
    "Tags": ["enc_base64", "active", "date20190923"],
    "Data": "test_4"
  },
  {
    "UUID": "fghj-98765-wdejpg-re4t5",
    "Tags": ["enc_base64", "active"],
    "Data": "test_3"
  }
]
```

نتایج به صورت زیر است.

The screenshot shows a Windows desktop with a VS Code editor open. The editor has a sidebar on the left with a file explorer showing a project structure. The main editor area displays a JavaScript file named `index.js` with the following code:

```

1 // ...
2 let Tags = JSON.stringify(['enc_base64', 'active']);
3
4 let SortIndex = 2;
5
6 let PageSize = 4;
7
8 let PageIndex = 3;
9
10 // ...

```

Below the editor, a terminal window is open, showing the output of a Node.js command. The terminal displays a series of log messages, including timestamps, IP addresses, and data objects. The messages are as follows:

```

PS C:\Users\SAM-Tech\Desktop\UdataDataChallenge_Ameri> node server.js
[2022-07-29T15:37:33.553] [INFO] default - [ 'enc_base64', 'active', 'date20130128' ]
[2022-07-29T15:37:33.554] [INFO] default - [ 'enc_base64', 'active', 'date20130109' ]
[2022-07-29T15:37:33.554] [INFO] default - [ 'enc_base64', 'active', 'date20130815' ]
[2022-07-29T15:37:33.555] [INFO] default - [ 'enc_base64', 'active', 'date20132218' ]
[2022-07-29T15:37:33.555] [INFO] default - [ 'enc_base64', 'active', 'date20130327' ]
[2022-07-29T15:37:33.555] [INFO] default - [ 'enc_base64', 'active', 'date20130414' ]
[2022-07-29T15:37:33.556] [INFO] default - [ 'enc_base64', 'active', 'date20130819' ]
[2022-07-29T15:37:33.556] [INFO] default - [ 'enc_base64', 'active', 'date20130107' ]
[2022-07-29T15:37:33.556] [INFO] default - [ 'enc_base64', 'active', 'date20130117' ]
[2022-07-29T15:37:33.557] [INFO] default - [ 'enc_base64', 'active', 'date20130823' ]
[2022-07-29T15:37:33.557] [INFO] default - [ 'enc_base64', 'active', 'date20130923' ]
[2022-07-29T15:37:33.557] [INFO] default - [ 'enc_base64', 'active', 'date20130923' ]
[2022-07-29T15:37:33.557] [INFO] default - [ 'enc_base64', 'active' ]
[2022-07-29T15:37:33.557] [INFO] default - [ 'enc_base64', 'active' ]
[2022-07-29T15:37:33.557] [INFO] default - length of Pages = 4
[2022-07-29T15:37:33.558] [INFO] default - [
  {
    UUID: 'f9b1-98765-wd6jg-re4t5',
    Tags: [ 'enc_base64', 'active' ],
    Data: 'test 1'
  },
  {
    UUID: 'f9b1-98765-wd6jg-re4t5',
    Tags: [ 'enc_base64', 'active' ],
    Data: 'test 3'
  }
]
PS C:\Users\SAM-Tech\Desktop\UdataDataChallenge_Ameri>

```

The terminal window also shows the command prompt and the file path `C:\Users\SAM-Tech\Desktop\UdataDataChallenge_Ameri`.

شکل ۱-۲- نتیجه تست کد index.js بر روی نمونه‌های بالا

server.js کد فایل

```
const http = require('http');
const log4js = require('log4js');
const fs = require('fs');

const logger = log4js.getLogger();
logger.level = 'debug';

const port = process.env.port || 5000;

const server = http.createServer((req, res) => {

  if (req.url == '/search') {
    const sample = fs.readFileSync('sample.json').toString();
    res.writeHead(200);
    res.write(sample);
    res.end();
  }
}).listen(port);
```

index.js کد فایل

```
const http = require('http');
const log4js = require('log4js');

const logger = log4js.getLogger();
logger.level = 'debug';

// let Tags = prompt("Enter Tags: ");
let Tags = JSON.stringify(['enc_base64', 'active']);
let SortIndex = 2;
let PageSize = 4;
let PageIndex = 6;

///// level 1
let optionSearch = {
  hostname: 'localhost',
  port: process.env.port || 5000,
  path: '/search',
  headers: {
    'Content-Type': 'application/json',
    'Content-Length': Tags.length
  },
  method: 'POST',
  timeout: 120000
};

let reqSearch = http.request(optionSearch, (res) => {
  if (res.statusCode == 200) {
    let DataSet = [];
    res.on('data', (chunk) => {
      DataSet = JSON.parse(chunk);
    });
  }
});
```

```

    res.on('end', () => {
        ///// Sorting
        // DataSet = DataSet.sort(a => a.Tags[SortIndex]);
        // logger.warn(DataSet);
        sort(DataSet, SortIndex);
        // quicksort(DataSet, 0, DataSet.length - 1);
        for (const data of DataSet) {
            try {
                logger.info(data.Tags);
            } catch (ex) {
                logger.info(data);
            }
        }

        ///// Paging
        let Pages = [];
        let len = Math.ceil(DataSet.length / PageSize);
        logger.info(`length of Pages = ${len}`)
        for (let i = 0; i < len; i++) {
            let Page = [];
            for (let j = 0; j < PageSize; j++) {
                let index = (i * PageSize) + j;
                let page = DataSet[index];
                if (page !== undefined)
                    Page[j] = page;
            }
            Pages[i] = Page;
        }
        if (len <= PageIndex)
            logger.error(`length of Pages is ${len} enter PageIndex`)
        else
            logger.info(Pages[PageIndex]);
    });
}

});
reqSearch.write(Tags);
reqSearch.end();

///// functions
function swap(array, i, j) {
    let temp = array[i];
    array[i] = array[j];
    array[j] = temp;
}

function sort(DataSet, SortIndex) {
    let undefines = [];
    for (let i = 0; i < DataSet.length; i++) {
        for (let j = 0; j < DataSet.length; j++) {
            try {
                if (DataSet[i].Tags[SortIndex] !== undefined) {
                    if (DataSet[i].Tags[SortIndex] <
DataSet[j].Tags[SortIndex])
                        swap(DataSet, i, j);
                } else {
                    swap(DataSet, i, DataSet.length - 1);
                    let temp = DataSet.pop();
                    undefines.push(temp);
                }
            } catch (ex) {
            }
        }
    }
}

```

```
for (const item of undefinedes) {  
    DataSet.push(item);  
}  
// logger.warn(undefinedes);  
}
```