به نام او که از ما به شما عاشق‌تر است

**تحلیل و طراحی چالش Data Date**

**گروه نرم افزاری پارت**

نفیسه عامری1

1 دانشجو 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.

# 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 پیاده سازی بخش اصلی برنامه، ابتدا پارامترهای Tags، SortIndex، PageSize و PageIndexاز کاربر دریافت می­کنیم. سپس متغییر 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 مرتب سازی و صفحه­بندی را با توجه به نیاز کاربر انجام می­دهیم.

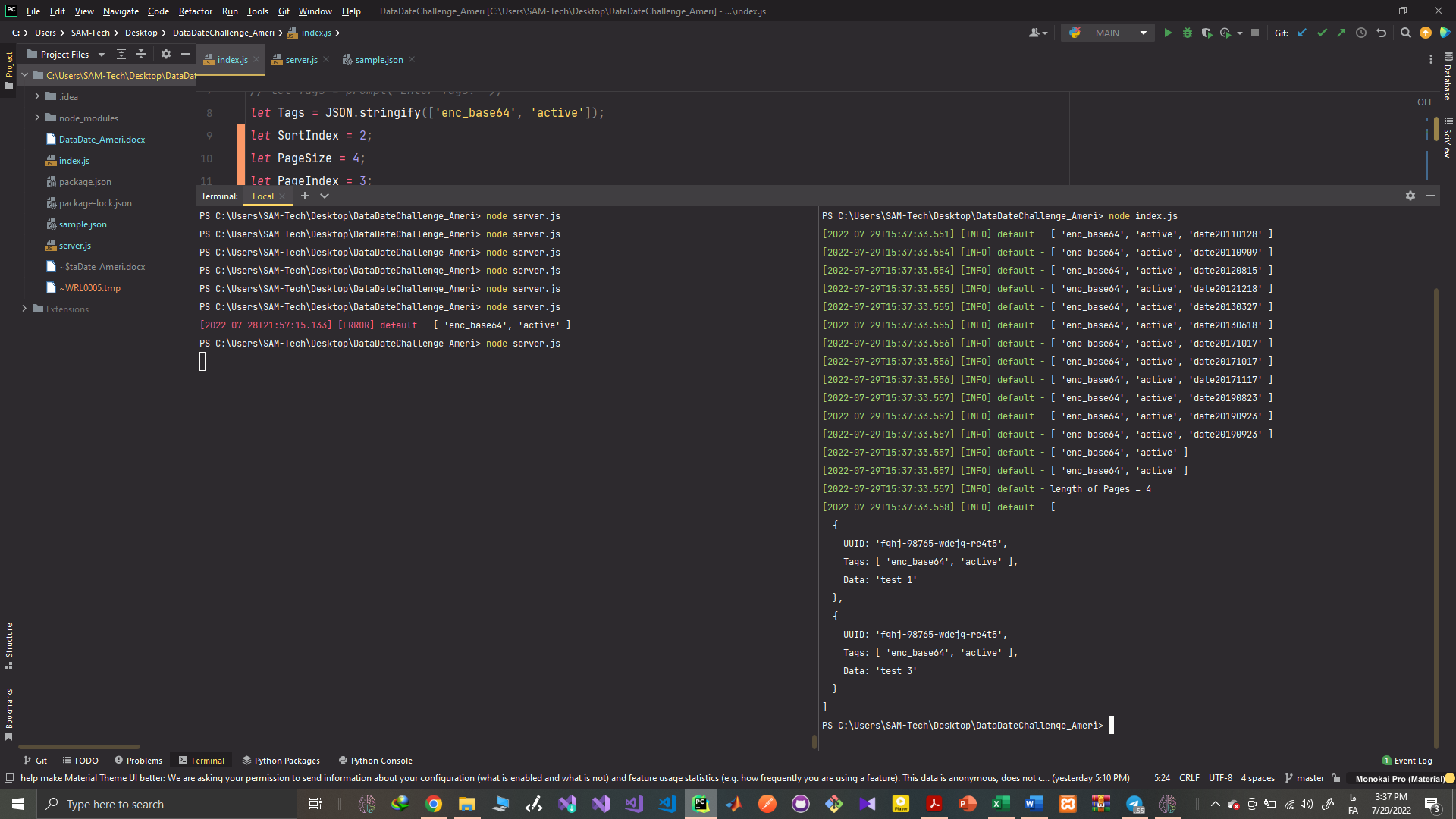
بر اساس مقدار SortIndex که از کاربر دریافت کردیم از لیست Tagsهای DataSet بر اساس اندیس مورد نظر مرتب سازی را انجام می­دهیم. بعد از مرتب سازی برای صفحه بندی آبجکت­ها از دو حلقه تودرتو استفاده می­کنیم و صفحه بندی را انجام می­دهیم. و در انتها باتوجه به مقدار PageIndex صفحه مورد نظر کاربر را برمیگردانیم.

# شرح نتایج

برای تست، از تعدادی نمونه زیر استفاده کردیم.

[  
 {  
 "UUID": "fghj-98765-wdejg-re4t5",  
 "Tags": ["enc\_base64", "active", "date20121218"],  
 "Data": "test 2"  
 },  
 {  
 "UUID": "fghj-98765-wdejg-re4t5",  
 "Tags": ["enc\_base64", "active", "date20130618"],  
 "Data": "test 4"  
 },  
 {  
 "UUID": "fghj-98765-wdejg-re4t5",  
 "Tags": ["enc\_base64", "active", "date20110128"],  
 "Data": "test 3"  
 },  
 {  
 "UUID": "fghj-98765-wdejg-re4t5",  
 "Tags": ["enc\_base64", "active", "date20190823"],  
 "Data": "test 4"  
 },  
 {  
 "UUID": "fghj-98765-wdejg-re4t5",  
 "Tags": ["enc\_base64", "active", "date20171017"],  
 "Data": "test 3"  
 },  
 {  
 "UUID": "fghj-98765-wdejg-re4t5",  
 "Tags": ["enc\_base64", "active", "date20120815"],  
 "Data": "test 1"  
 },  
 {  
 "UUID": "fghj-98765-wdejg-re4t5",  
 "Tags": ["enc\_base64", "active", "date20130327"],  
 "Data": "test 4"  
 },  
 {  
 "UUID": "fghj-98765-wdejg-re4t5",  
 "Tags": ["enc\_base64", "active", "date20190923"],  
 "Data": "test 4"  
 },  
 {  
 "UUID": "fghj-98765-wdejg-re4t5",  
 "Tags": ["enc\_base64", "active", "date20171117"],  
 "Data": "test 3"  
 },  
 {  
 "UUID": "fghj-98765-wdejg-re4t5",  
 "Tags": ["enc\_base64", "active", "date20171017"],  
 "Data": "test 3"  
 },  
 {  
 "UUID": "fghj-98765-wdejg-re4t5",  
 "Tags": ["enc\_base64", "active"],  
 "Data": "test 1"  
 },  
 {  
 "UUID": "fghj-98765-wdejg-re4t5",  
 "Tags": ["enc\_base64", "active", "date20110909"],  
 "Data": "test 5"  
 },  
 {  
 "UUID": "fghj-98765-wdejg-re4t5",  
 "Tags": ["enc\_base64", "active", "date20190923"],  
 "Data": "test 4"  
 },  
 {  
 "UUID": "fghj-98765-wdejg-re4t5",  
 "Tags": ["enc\_base64", "active"],  
 "Data": "test 3"  
 }  
]

نتایج به صورت زیر است.



شکل ‏2‑1-نتیجه تست کد index.js بر روی نمونه­های بالا

**پیوست**

**کد فایل server.js**

*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);

**کد فایل index.js**

*const* http = require('http');  
*const* log4js = require('log4js');  
  
*const* logger = log4js.getLogger();  
logger.level = 'debug';  
  
 *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* undefinedes = [];  
 *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();  
 undefinedes.push(temp);  
 }  
 } *catch* (ex) {  
 }  
 }  
 }  
 *for* (*const* item *of* undefinedes) {  
 DataSet.push(item);  
 }  
 *// logger.warn(undefinedes);*}