مکتب شریف

اولین بوتکمپ آموزشی - استخدامی ایران

مکتب 137

سری دوم

سوال ۱: پردازشگر فایل لاگ سرور (Server Log Processor)

شما یک فایل لاگ بزرگ به نام server.log در اختیار دارید. هر خط از این فایل یک رویداد را ثبت می‌کند که شامل تاریخ، سطح لاگ (INFO, WARNING, ERROR) و یک پیام است. متأسفانه، برخی از خطوط این فایل ممکن است به دلیل خطا در سیستم، فرمت درستی نداشته باشند.

وظایف:

1. تابعی بنویسید که فایل لاگ را به صورت خط به خط بخواند تا در کار با فایل‌های حجیم، حافظه زیادی مصرف نشود.

2. برای هر خط، اطلاعات level ,timestamp, و message را استخراج کنید. فرمت هر خط معتبر به این صورت است:

```
[Day Abbr Month DD HH:MM:SS YYYY] [LEVEL] Message
```

برای دانلود فایل لاگ اینجا کلیک کنید

3. اگر یک خط فرمت درستی نداشت، آن را نادیده بگیرید، اما شماره آن خط و محتوای آن را در یک فایل جداگانه به نام errors.log ذخیره کنید.

4. لاگ‌های معتبر را دسته‌بندی کنید:
   - تمام لاگ‌هایی که سطح آن‌ها ERROR است را در یک فایل CSV به نام critical_errors.csv با ستون‌های Timestamp, Message ذخیره کنید.

5. در پایان، تعداد کل لاگ‌های هر سطح (INFO, WARNING, ERROR) را شمرده و نتیجه را در یک فایل JSON به نام summary.json به این صورت ذخیره کنید:

```
}
INFO": 1024"  ,
WARNING": 56"  ,
ERROR": 12"
{
```

6. کل فرآیند باید داخل یک بلوک try-except قرار گیرد تا خطای FileNotFoundError (در صورتی که فایل server.log وجود نداشته باشد) را مدیریت کرده و پیام مناسبی به کاربر نمایش دهد.

سوال ۲: مدیر پیکربندی برنامه (Application Configuration Manager)

یک برنامه برای تنظیمات خود از یک فایل config.json استفاده می‌کند. این تنظیمات شامل اطلاعات اتصال به پایگاه داده، کلیدهای API و یک فیلد mode است که می‌تواند debug یا production باشد.

وظایف:

1. یک کلاس استثناء (Exception) سفارشی به نام `InvalidConfigError` ایجاد کنید.

2. تابعی به نام `load_config(filepath)` بنویسید که فایل پیکربندی JSON را می‌خواند.

3. این تابع باید موارد زیر را اعتبارسنجی کند:
   - اگر فایل وجود نداشت، باید خطای `FileNotFoundError` را صادر کند.
   - اگر محتوای فایل، یک JSON معتبر نبود، باید خطای `JSONDecodeError` را مدیریت کند.
   - باید بررسی کند که کلیدهای ضروری `'api_key'`, `'database_url'`, و `'mode'` در فایل وجود داشته باشند. اگر هرکدام از آن‌ها موجود نبود، باید یک استثناء از نوع `InvalidConfigError` با پیام مناسب صادر کند (raise).
   - باید بررسی کند که مقدار کلید `'mode'` حتماً یکی از دو مقدار `'debug'` یا `'production'` باشد. در غیر این صورت، یک `InvalidConfigError` با پیام مناسب صادر کند.

4. در بخش اصلی برنامه، تابع load_config را فراخوانی کنید و با استفاده از بلوک‌های try-except تمام خطاهای احتمالی (نبودن فایل، نامعتبر بودن JSON، و خطاهای پیکربندی) را به شکل مناسبی مدیریت کرده و پیام‌های واضح و کاربرپسندی را چاپ کنید.

سوال ۳: پایپ‌لاین کوچک داده (Mini Data-Pipeline)

❖ این سوال دارای نمره اضافه می باشد. (extra point)

شما باید ابزار کوچکی بسازید که اطلاعات کاربران را از یک API عمومی دریافت کرده، آن‌ها را پردازش و در نهایت ذخیره کند.

وظایف:

1. یک ماژول پایتون به نام api_client.py ایجاد کنید. در این ماژول، تابعی به نام fetch_data(api_url) بنویسید که با استفاده از کتابخانه requests اطلاعات را از URL داده شده دریافت می‌کند. این تابع باید خطاهای احتمالی شبکه (مانند requests.exceptions.RequestException) را مدیریت کرده و در صورت موفقیت، داده‌های JSON (لیستی از دیکشنری‌های کاربران) را برگرداند و در صورت بروز خطا، None را بازگرداند.

2. یک ماژول دیگر به نام data_processor.py ایجاد کنید. در این ماژول، تابعی به نام process_and_save(users_data, output_path) بنویسید. این تابع لیستی از دیکشنری‌های کاربران را به عنوان ورودی می‌گیرد.

3. تابع process_and_save باید کارهای زیر را انجام دهد:
   - کاربران را فیلتر کند (برای مثال، فقط کاربرانی که کد پستی آن‌ها با عدد '9' شروع می‌شود را نگه دارد).
   - برای کاربران فیلتر شده، فقط اطلاعات 'email', 'name', و 'company name' را استخراج کند.
   - در نهایت، این اطلاعات پردازش شده را در یک فایل CSV در مسیر output_path ذخیره کند.

4. یک فایل اصلی به نام main.py ایجاد کنید که توابع را از دو ماژول دیگر import کند.
   - در این فایل، تابع fetch_data را با یک API عمومی (مانند https://jsonplaceholder.typicode.com/users) فراخوانی کنید.
   - اگر داده‌ها با موفقیت دریافت شدند، آن‌ها را به تابع process_and_save ارسال کنید تا پردازش و ذخیره شوند.
   - برنامه اصلی باید حالتی که fetch_data مقدار None برمی‌گرداند را نیز مدیریت کند و پیام مناسبی نمایش دهد.

نکات :

- مهلت ارسال تمرین تا پایان روز چهارشنبه 02 / 07 / 1404 است
- پاسخ تمرین را در سامانه مودل ارسال کنید
- نام فایل ارسالی خود را به این صورت قرار دهید : Name_hw1_maktab137

به عنوان مثال Mahdi_Zolfaghari_hw1_maktab137

- در صورتی که تمرین شامل چند فایل یا فولدر میباشد حتما آن را در قالب یک فایل فشرده شده تجمیع کنید.