task

각 크롤러의 람다 코드에서 함수별 진행도와 예외처리 부분에 로그를 남기는 코드를
추가해주십시오.

1. cloudwatch_handler.py 파일을 생성해줍니다.

```python
import logging
from botocore.exceptions import NoCredentialsError,
PartialCredentialsError
# CloudWatch Logs 클라이언트 생성
# client = boto3.client('logs', region_name='ap-northeast-2')  # 서울
리전
import json, boto3

client = boto3.client('logs')

class cloudwatch_handler(logging.Handler):
    log_group_name = None
    log_stream_name = None

    def set_init(self,group_name,stream_name):
        self.log_group_name = group_name
        self.log_stream_name = stream_name
        try:
            client.create_log_group(logGroupName=group_name)
        except client.exceptions.ResourceAlreadyExistsException:
            pass
        try:
            client.create_log_stream(logGroupName=group_name,
logStreamName=stream_name)
        except client.exceptions.ResourceAlreadyExistsException:
            pass

    def emit(self, record):
        log_entry = self.format(record)
        try:
            response = client.put_log_events(
                logGroupName=self.log_group_name,
                logStreamName=self.log_stream_name,
                logEvents=[
                    {
                        'timestamp': int(record.created * 1000),
                        'message': log_entry
                    },
```

```
                    ],
                )
        except (NoCredentialsError, PartialCredentialsError) as e:
            print(f"Credentials error: {e}")
        except Exception as e:
            print(f"Error sending log to CloudWatch: {e}")
```

2. logging_to_cloudwatch.py 파일을 생성해줍니다.

```python
import logging
import cloudwatch_handler

def log(path, stream_name):
    # CloudWatchHandler 생성
    handler = cloudwatch_handler.cloudwatch_handler()
    # 로깅 설정
    # logging.basicConfig(level=logging.INFO)

    logger = logging.getLogger(__name__)
    logger.setLevel(logging.DEBUG)
    handler.set_init(path, stream_name)
    handler.setLevel(logging.DEBUG)
    formatter = logging.Formatter("[%(levelname)s] %(message)s",
datefmt="%Y-%m-%d %H:%M:%S")
    handler.setFormatter(formatter)
    logger.addHandler(handler)

    return logger
```

3. 아래코드를 각 크롤러의 메인 파일에 추가해주시고 함수에 로그를 추가해주십시오

```python
import logging
import logging_to_cloudwatch as ltc


logger = ltc.log('/aws/lambda/{cloudwatch의
크롤러폴더}','{크롤러명}_logs')
```

예시)

```python
def company_code(payload):
    logger.info("Collecting company_code start")
...
```

```
    logger.info("Collecting company_code done")
    return company_num


def job_id(payload):
    logger.info("Collecting job_id start")
...
    logger.info("Collecting job_id done")
    return jobid



def makedf(payload, crawl_time):
    logger.info("Collecting make_dataframe start")

...
    except Exception as e:
        logger.error(f"An error occurred while collecting data from
{job_url}, error: {str(e)} , continue")
        continue
```

========================
실행결과

[INFO] Collecting make_dataframe start

[INFO] Collecting job_id start

[INFO] Collecting company_code start

[INFO] Collecting company_code done

[INFO] Collecting job_id done

[ERROR] An error occurred while collecting data from https://career.programmers.co.kr/api/job_positions/15521, error: 'jobPosition'

[ERROR] An error occurred while collecting data from https://career.programmers.co.kr/api/job_positions/15249, error: 'jobPosition'

[ERROR] An error occurred while collecting data from https://career.programmers.co.kr/api/job_positions/15250, error: 'jobPosition'

[ERROR] An error occurred while collecting data from https://career.programmers.co.kr/api/job_positions/15261, error: 'jobPosition'

[ERROR] An error occurred while collecting data from https://career.programmers.co.kr/api/job_positions/15258, error: 'jobPosition'

[ERROR] Error loading offset: name 'crawl_time' is not defined