IITP DABT PreProcessing

설계서

문서 버전: 1.0.0
작성일: 2025-11-25

(주)스위트케이

문서 History

버전	일자	작성자	변경 내용
1.0.0	2025-11-25	(주)스위트케이	최초 작성

목차

1. 개요

1.1. 프로젝트 소개

IITP DABT PreProcessing은 KOSIS(국가통계포털)의 공개 API를 통해 통계 데이터를 수집하고, 이를 파일 시스템에 저장하며 필요시 데이터베이스에 적재하는 전처리 시스템입니다.

본 시스템은 Python 기반으로 개발되었으며, 대용량 통계 데이터의 효율적인 수집 및 관리와 데이터 일관성 보장을 목적으로 합니다. 시스템은 배치 처리 방식으로 동작하며, 병렬 처리를 통해 성능을 최적화합니다.

1.2. 핵심 기능

- **외부 API 연동**: KOSIS API를 통한 통계 데이터, 메타데이터, 최신 변경일자 정보 수집
- **파일 기반 저장**: 수집된 데이터를 날짜별 폴더 구조로 체계적으로 관리
- **데이터베이스 적재**: 원본 데이터, 메타데이터, 통합 테이블로의 자동 이관
- **병렬 처리**: 다중 통계 데이터의 동시 수집 및 처리로 성능 향상
- **데이터 정합성 관리**: 트랜잭션 기반 처리 및 과거 데이터 자동 정리
- **에러 복구**: 대용량 데이터 수집 시 자동 분할 처리 및 에러 핸들링
- **유연한 실행 모드**: 파일 저장 전용 모드 및 DB 삽입 모드 지원

2. 프로젝트 전체 소스 구조 및 설명

2.1. 디렉토리 구조

프로젝트의 디렉토리 구조는 다음과 같습니다:

```
프로젝트 루트/
├── main.py                 # 메인 진입점 및 전체 프로세스 제어
├── db.py                   # 데이터베이스 연결 및 조회 모듈
├── db_processing.py        # 데이터베이스 삽입 및 처리 로직
├── kosis_api.py            # KOSIS API 호출 및 데이터 수집
├── file_utils.py           # 파일 저장 유틸리티
├── config.py               # 환경 설정 및 구성 관리
├── requirements.txt        # Python 패키지 의존성 목록
├── README.md               # 프로젝트 문서
├── logs/                   # 실행 로그 저장 디렉토리
│   ├── YYYYMMDD.log        # 일반 실행 로그
│   └── db_YYYYMMDD.log     # DB 처리 전용 로그
└── kosis_data/             # 수집된 데이터 저장 디렉토리
    └── YYYYMMDD/           # 실행일자별 폴더
        ├── data/           # 통계 데이터 파일 (JSON)
        ├── meta/           # 메타데이터 파일 (XML/JSON)
        └── latest/         # 최신 변경일자 정보 파일 (XML/JSON)
```

2.2. 주요 모듈 설명

2.2.1. 메인 모듈 (main.py)

시스템의 진입점으로 전체 프로세스를 제어합니다. 주요 역할은 다음과 같습니다:

- 명령행 인자 파싱 및 검증
- 로깅 시스템 초기화
- 데이터 수집 범위 설정 및 필터링
- 디렉토리 구조 생성
- 파일 저장 프로세스 조율
- DB 삽입 모드 선택적 실행

2.2.2. 데이터베이스 모듈 (db.py)

데이터베이스 연결 및 조회 기능을 제공합니다:

- SQLAlchemy 엔진 및 세션 관리
- 타임존 설정 (Asia/Seoul)
- 외부 API 정보 조회
- 통계 소스 API 정보 조회
- 통계 소스 데이터 정보 조회

2.2.3. DB 처리 모듈 (db_processing.py)

데이터베이스에 데이터를 삽입하고 관리하는 핵심 로직을 담당합니다:

- 원본 데이터 테이블 삽입
- 통합 테이블로의 데이터 이관
- 메타데이터 저장
- 통계 소스 정보 업데이트
- 시스템 데이터 요약 정보 업데이트
- 과거 데이터 정리

2.2.4. API 호출 모듈 (kosis_api.py)

KOSIS API와의 통신을 담당합니다:

- API URL 구성 및 파라미터 치환
- 데이터 수집 (에러 31 자동 분할 처리)
- 메타데이터 수집
- 최신 변경일자 정보 수집
- 에러 처리 및 재시도 로직

2.2.5. 파일 유틸리티 모듈 (file_utils.py)

파일 저장 관련 기능을 제공합니다:

- 안전한 파일명 생성 (특수문자 처리)
- 데이터 파일 저장
- 메타데이터 파일 저장
- 최신 변경일자 파일 저장
- 파일명 규칙 적용

2.2.6. 설정 모듈 (config.py)

환경 변수 및 설정값을 관리합니다:

- 환경 변수 로드 (.env 파일)
- 데이터베이스 연결 정보
- 로그 레벨 설정
- 병렬 처리 워커 수 설정
- 배치 크기 설정
- 데이터 수집 범위 설정

2.3. 프로젝트 의존성

프로젝트에서 사용하는 주요 외부 패키지는 다음과 같습니다:

패키지명	용도	버전
requests	HTTP API 호출	>=2.32.4
python-dotenv	환경변수 관리	>=1.0.1
sqlalchemy	DB ORM	>=2.0.30
psycopg2-binary	PostgreSQL 드라이버	>=2.9.10

2.3.1. 의존성 구조

- **HTTP 통신**: requests 라이브러리를 사용하여 KOSIS API와 통신
- **데이터베이스**: SQLAlchemy ORM을 통해 PostgreSQL과 연동
- **설정 관리**: python-dotenv를 통해 환경 변수 관리
- **병렬 처리**: Python 표준 라이브러리의 concurrent.futures 사용

2.4. 패키지 구조

시스템은 모듈화된 구조로 설계되어 각 모듈이 명확한 책임을 가집니다:

- **계층 구조**:
  - 진입점 계층 (main.py)
  - 비즈니스 로직 계층 (db_processing.py, kosis_api.py)
  - 유틸리티 계층 (file_utils.py, config.py)

    - 데이터 접근 계층 (db.py)
- **의존성 방향**: 상위 계층에서 하위 계층으로의 단방향 의존성 유지

2.5. 환경 설정 및 구성 관리

2.5.1. 환경 변수 설정

시스템은 `.env` 파일을 통해 환경 변수를 관리합니다. 주요 환경 변수는 다음과 같습니다:

데이터베이스 설정

- **DB_URL**: PostgreSQL 데이터베이스 연결 문자열
    - 형식: `postgresql://사용자명:비밀번호@호스트:포트/데이터베이스명`

로깅 설정

- **LOG_LEVEL**: 로그 레벨 설정 (DEBUG, INFO, WARNING, ERROR)
    - 기본값: INFO

성능 설정

- **DB_BATCH_SIZE**: 데이터베이스 배치 삽입 크기
    - 기본값: 100
    - 설명: 한 번에 삽입할 레코드 수
- **PARALLEL_WORKERS_FILE**: 파일 저장 병렬 처리 워커 수
    - 기본값: 4
    - 최대값: 10 (자동 제한)
- **PARALLEL_WORKERS_DB**: DB 처리 병렬 처리 워커 수
    - 기본값: 2
    - 최대값: 5 (자동 제한)

데이터 수집 설정

- **EXT_API_INFO_KOSIS_SYS**: KOSIS 시스템 식별자
    - 기본값: KOSIS
- **DATA_COLLECTION_SCOPE**: 데이터 수집 범위
    - ALL: 모든 통계 데이터 수집
    - PART: 지정된 통계만 수집
    - 기본값: ALL
- **TARGET_SRC_TBL_ID_LIST**: PART 모드 시 수집 대상 통계 목록

- 형식: `[TARGET_SRC_TBL_ID_LIST]` 섹션에 `stat_tbl_id,from_year` 형식으로 나열
- 예시:

```
[TARGET_SRC_TBL_ID_LIST]
STAT001,2019
STAT002,2020
```

2.5.2. 설정 파일 구조

설정은 다음 순서로 우선순위를 가집니다:

1. 환경 변수 (시스템 환경 변수)
2. `.env` 파일
3. 기본값 (config.py에 하드코딩된 값)

2.5.3. 주요 설정 항목별 상세 설명

데이터베이스 배치 크기 (DB_BATCH_SIZE)

대량의 데이터를 삽입할 때 한 번에 처리할 레코드 수를 결정합니다. 값이 클수록:

- 장점: 삽입 성능 향상
- 단점: 메모리 사용량 증가, 트랜잭션 롤백 시 영향 범위 확대

권장값: 100~500 (데이터 크기 및 서버 성능에 따라 조정)

병렬 처리 워커 수

파일 저장 워커 (PARALLEL_WORKERS_FILE)

- 여러 통계 데이터를 동시에 수집 및 저장할 때 사용
- API 호출 제한 및 네트워크 대역폭을 고려하여 설정
- 권장값: 4~8

DB 처리 워커 (PARALLEL_WORKERS_DB)

- 여러 통계 데이터를 동시에 DB에 삽입할 때 사용
- 데이터베이스 연결 풀 및 트랜잭션 격리 수준을 고려하여 설정
- 권장값: 2~4

데이터 수집 범위 (DATA_COLLECTION_SCOPE)

ALL 모드

- 데이터베이스에 등록된 모든 활성화된 통계 데이터를 수집
- 전체 데이터 동기화 시 사용

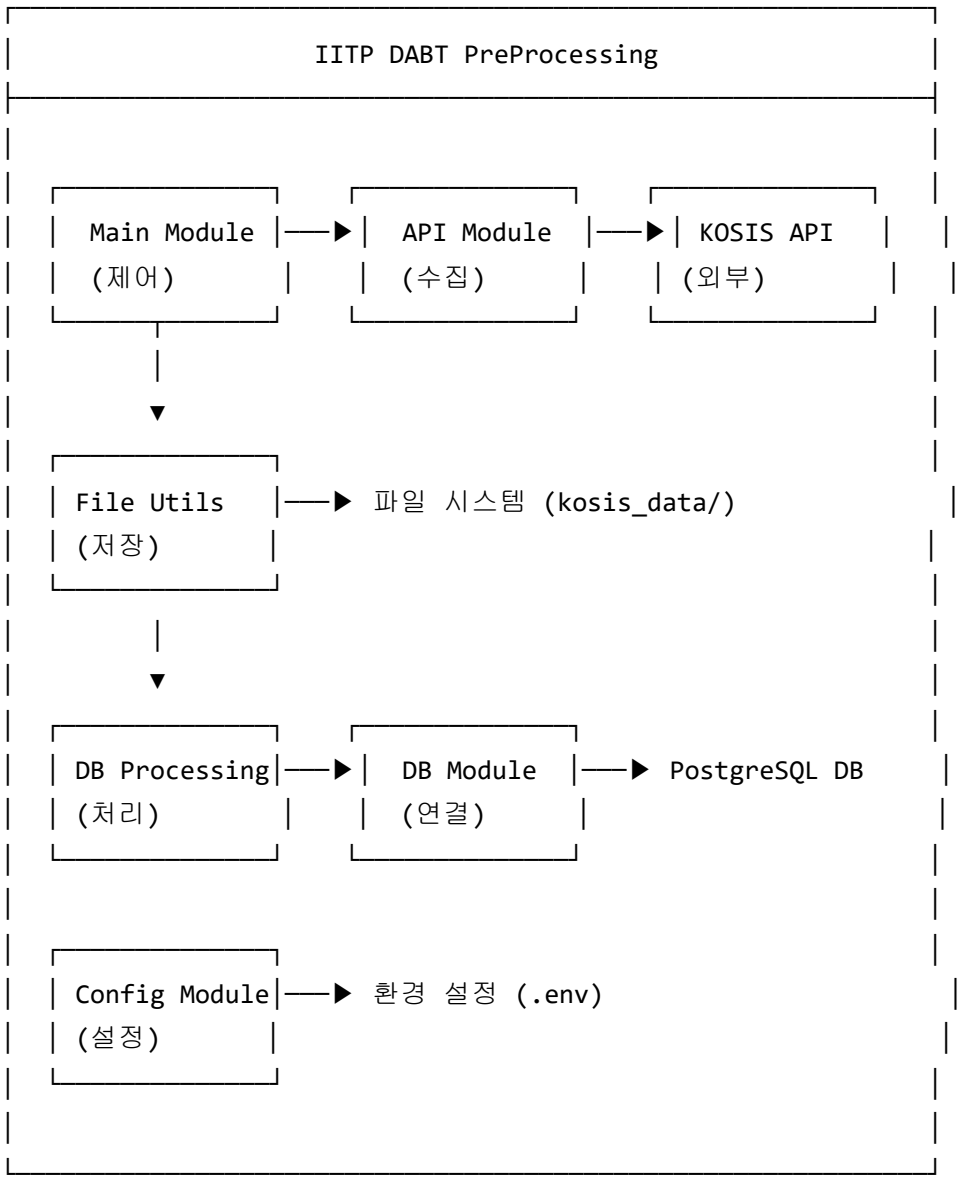
PART 모드

- 특정 통계만 선별적으로 수집
- 개발/테스트 환경 또는 특정 통계만 업데이트할 때 사용
- TARGET_SRC_TBL_ID_LIST에 명시된 통계만 처리

3. 프로젝트의 소프트웨어 아키텍처

3.1. 시스템 구성도

시스템은 다음과 같은 구성 요소로 이루어져 있습니다:



3.2. 아키텍처 설계 원칙

3.2.1. 모듈화

각 기능을 독립적인 모듈로 분리하여 유지보수성과 재사용성을 높였습니다.

3.2.2. 단일 책임 원칙

각 모듈은 하나의 명확한 책임만을 가지도록 설계되었습니다.

3.2.3. 의존성 역전

상위 계층이 하위 계층에 의존하되, 인터페이스를 통해 결합도를 낮췄습니다.

3.2.4. 확장성

새로운 API나 데이터 소스를 추가하기 쉽도록 확장 가능한 구조로 설계되었습니다.

3.3. 주요 컴포넌트

3.3.1. 제어 컴포넌트

- **역할**: 전체 프로세스의 흐름 제어
- **위치**: main.py
- **주요 기능**: 초기화, 프로세스 조율, 에러 처리

3.3.2. 데이터 수집 컴포넌트

- **역할**: 외부 API로부터 데이터 수집
- **위치**: kosis_api.py
- **주요 기능**: API 호출, 에러 처리, 자동 분할 수집

3.3.3. 데이터 저장 컴포넌트

- **역할**: 수집된 데이터를 파일 시스템에 저장
- **위치**: file_utils.py
- **주요 기능**: 파일명 생성, 데이터 저장, 디렉토리 관리

3.3.4. 데이터 처리 컴포넌트

- **역할**: 데이터베이스에 데이터 삽입 및 관리
- **위치**: db_processing.py
- **주요 기능**: 데이터 변환, 테이블 삽입, 트랜잭션 관리

3.3.5. 데이터 접근 컴포넌트

- **역할**: 데이터베이스 연결 및 조회
- **위치**: db.py
- **주요 기능**: 연결 관리, 쿼리 실행, 세션 관리

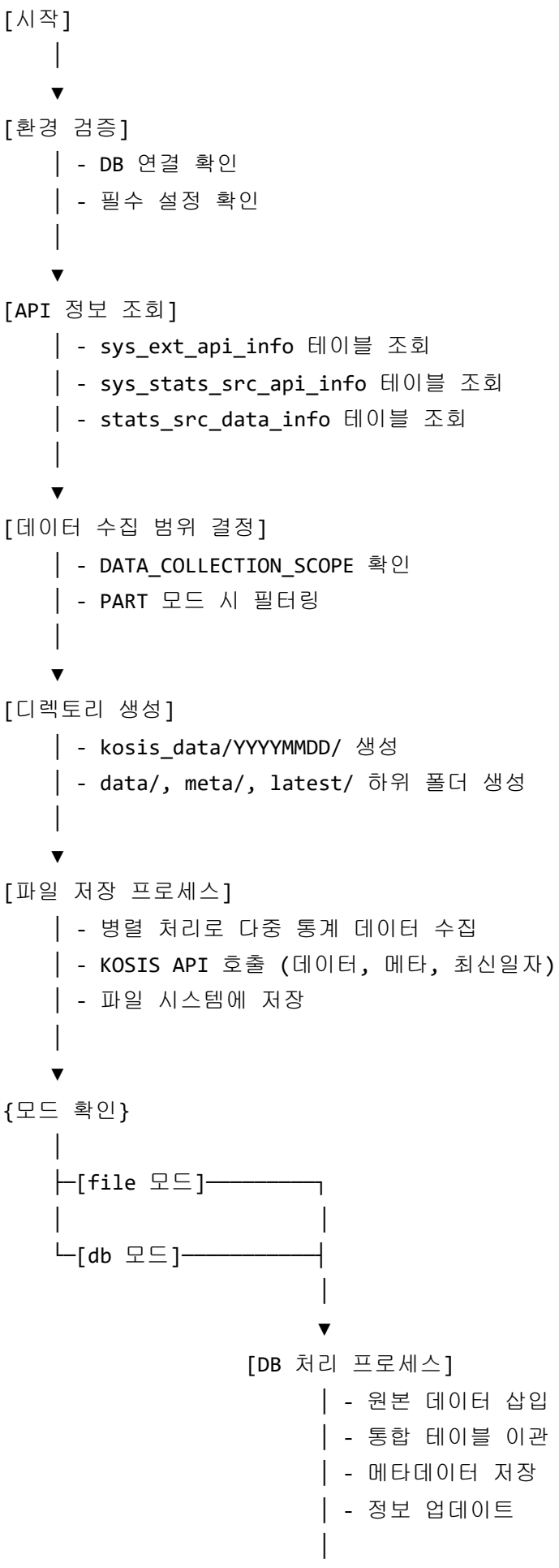
3.3.6. 설정 관리 컴포넌트

- **역할**: 시스템 설정 및 환경 변수 관리
- **위치**: config.py
- **주요 기능**: 설정 로드, 기본값 제공, 검증

4. 전체 시스템 연동 Flow

4.1. 전체 처리 흐름도

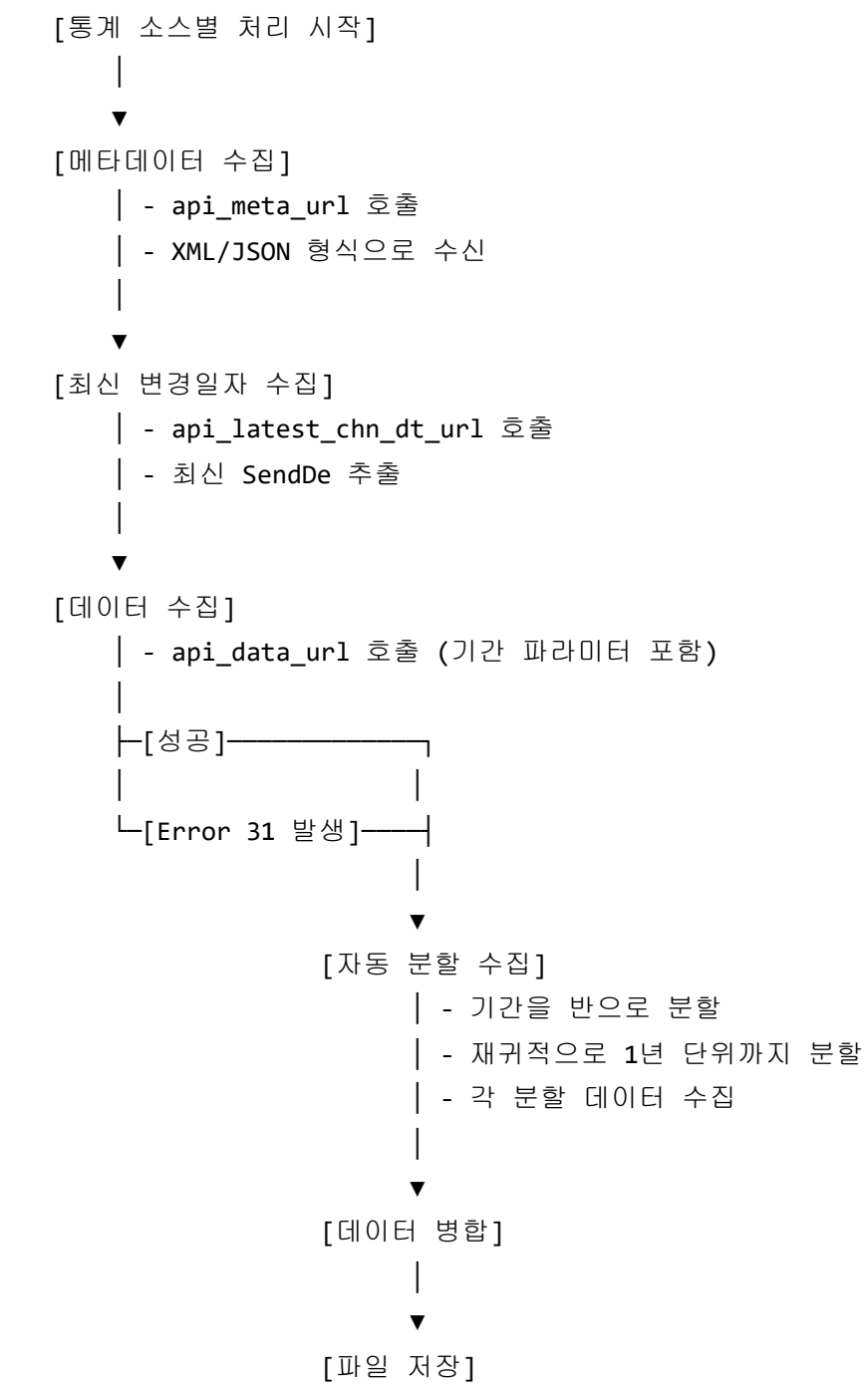
시스템의 전체 처리 흐름은 다음과 같습니다:


[시작]
[환경 검증]
- DB 연결 확인
- 필수 설정 확인
[API 정보 조회]
- sys_ext_api_info 테이블 조회
- sys_stats_src_api_info 테이블 조회
- stats_src_data_info 테이블 조회
[데이터 수집 범위 결정]
- DATA_COLLECTION_SCOPE 확인
- PART 모드 시 필터링
[디렉토리 생성]
- kosis_data/YYYYMMDD/ 생성
- data/, meta/, latest/ 하위 폴더 생성
[파일 저장 프로세스]
- 병렬 처리로 다중 통계 데이터 수집
- KOSIS API 호출 (데이터, 메타, 최신일자)
- 파일 시스템에 저장
{모드 확인}
[file 모드]
[db 모드]
[DB 처리 프로세스]
- 원본 데이터 삽입
- 통합 테이블 이관
- 메타데이터 저장
- 정보 업데이트

```
      ▼
[과거 데이터 정리]
      |
      ▼
[동기화 정보 업데이트]
      |
      ▼
[완료]
```

4.2. 데이터 수집 Flow

데이터 수집 프로세스의 상세 흐름은 다음과 같습니다:



4.3. 파일 저장 Flow

파일 저장 프로세스는 다음과 같이 진행됩니다:

```
[파일 저장 요청]
    │
    ▼
[파일명 생성]
    │ - 통계 제목, 기간, 타임스탬프 조합
    │ - 특수문자 제거/치환
    │ - 최대 길이 제한
    │
    ▼
[파일 형식 결정]
    │ - JSON 또는 XML
    │ - API 응답 형식에 따라 결정
    │
    ▼
[디렉토리 선택]
    │ - data/: 통계 데이터
    │ - meta/: 메타데이터
    │ - latest/: 최신 변경일자
    │
    ▼
[파일 저장]
    │ - UTF-8 인코딩
    │ - JSON: indent=2 포맷팅
    │ - XML: 원본 텍스트 저장
    │
    ▼
[저장 경로 반환]
```

4.4. DB 처리 Flow

데이터베이스 처리 프로세스는 다음 단계로 구성됩니다:

```
[DB 처리 시작]
    │
    ▼
[통계별 병렬 처리]
    │ - ThreadPoolExecutor 사용
    │ - 각 통계마다 독립 세션
    │
    ▼
[단일 통계 처리]
    │
    ├─[1단계: 최신일자 추출]
    │   │ - latest XML 파일 파싱
    │   │ - SendDe 중 최신값 추출
    │   │
    ├─[2단계: 원본 데이터 삽입]
    │   │ - data JSON 파일 로드
    │   │ - stats_kosis_origin_data 테이블에 bulk insert
    │   │ - 배치 단위로 처리
    │   │
    ├─[3단계: 통합 테이블 이관]
    │   │ - 기존 데이터 삭제 (동일 날짜, 오늘 이전)
    │   │ - 원본 데이터에서 SELECT하여 통합 테이블에 INSERT
    │   │ - 데이터 타입 변환 (문자열→숫자 등)
    │   │
    ├─[4단계: 메타데이터 저장]
    │   │ - 기존 메타데이터 삭제
    │   │ - meta XML 파일 파싱
    │   │ - stats_kosis_metadata_code 테이블에 bulk insert
    │   │
    ├─[5단계: 통계 소스 정보 업데이트]
    │   │ - stats_src_data_info 테이블 업데이트
    │   │ - stat_latest_chn_dt, stat_data_ref_dt 갱신
    │   │ - avail_cat_cols 업데이트
    │   │
    ├─[6단계: 시스템 데이터 요약 업데이트]
    │   │ - sys_data_summary_info 테이블 업데이트
    │   │ - src_latest_chn_dt, sys_data_ref_dt 갱신
    │   │
    └─[7단계: 관리 테이블 업데이트]
        │ - sys_stats_src_api_info 테이블 업데이트
        │ - latest_sync_time 갱신
        │
        ▼
```

```
    [트랜잭션 커밋]
        |
        ▼
[모든 통계 처리 완료]
    |
    ▼
[전체 동기화 정보 업데이트]
    | - sys_ext_api_info 테이블 업데이트
    |
    ▼
[과거 데이터 정리]
    | - 최신 데이터만 유지
    | - 오늘 생성된 데이터 제외하고 삭제
    |
    ▼
[완료]
```

5. 전체 시스템의 기능 상세 설명

5.1. 데이터 수집 기능

5.1.1. API 정보 조회

시스템은 데이터베이스에서 KOSIS API 연동 정보를 조회합니다.

참조 테이블:

- `sys_ext_api_info` : 외부 API 기본 정보 (API ID, URL, 인증키 등)
- `sys_stats_src_api_info` : 통계별 API URL 정보 (데이터 URL, 메타 URL, 최신일자 URL)
- `stats_src_data_info` : 통계 데이터 정보 (수집 기간, 통계 제목 등)

처리 과정:

1. 환경 변수에서 KOSIS 시스템 식별자 조회
2. `sys_ext_api_info` 테이블에서 활성화된 API 정보 조회
3. 해당 API ID로 `sys_stats_src_api_info` 테이블에서 통계 목록 조회
4. 각 통계의 `stat_tbl_id` 로 `stats_src_data_info` 테이블에서 상세 정보 조회

5.1.2. 데이터 수집

KOSIS API를 호출하여 통계 데이터를 수집합니다.

주요 기능:

- **데이터 수집**: 기간 파라미터를 포함한 API 호출
- **메타데이터 수집**: 통계의 분류체계 및 코드 정보 수집
- **최신 변경일자 수집**: 데이터의 최신 갱신일자 정보 수집

에러 처리:

- **Error 31 처리**: 대용량 데이터 수집 시 발생하는 에러를 자동으로 처리
  - 전체 기간 수집 실패 시 기간을 반으로 분할
  - 1년 단위까지 재귀적으로 분할하여 수집
  - 각 분할 데이터를 병합하여 반환

병렬 처리:

- 여러 통계 데이터를 동시에 수집하여 전체 처리 시간 단축
- ThreadPoolExecutor를 사용한 멀티스레드 처리
- 워커 수는 환경 변수로 설정 가능

5.1.3. URL 구성

API URL은 다음과 같이 구성됩니다:

- **Base URL**: `sys_ext_api_info.ext_url` (옵션)
- **API URL**: `sys_stats_src_api_info.api_data_url` (JSON 형식)
- **파라미터 치환**:
  - `{API_AUTH_KEY}` → `sys_ext_api_info.auth`
  - `{from}` → 수집 시작 연도
  - `{to}` → 수집 종료 연도

5.2. 파일 저장 기능

5.2.1. 디렉토리 구조 생성

실행일자를 기준으로 다음과 같은 디렉토리 구조를 생성합니다:

```
kosis_data/
└── YYYYMMDD/          # 실행일자
    ├── data/          # 통계 데이터 파일
    ├── meta/          # 메타데이터 파일
    └── latest/        # 최신 변경일자 파일
```

5.2.2. 파일명 규칙

파일명은 다음 규칙에 따라 생성됩니다:

- **데이터 파일**: `data_{src_data_id}-{stat_title}-{from_year}-{to_year}_{timestamp}.json`
- **메타데이터 파일**: `meta_{src_data_id}-{stat_title}-{from_year}-{to_year}_{timestamp}.xml`
- **최신일자 파일**: `latest_{src_data_id}-{stat_title}-{from_year}-{to_year}_{timestamp}.xml`

파일명 처리:

- 특수문자 (`\` , `/` , `:` , `*` , `?` , `"` , `<` , `>` , `|`) 제거 또는 언더스코어로 치환
- 최대 길이 제한 (기본 100자)
- UTF-8 인코딩 사용

5.2.3. 파일 저장 형식

- **JSON 파일**: `json.dump()` 사용, `indent=2` 로 포맷팅, `ensure_ascii=False` 로 한글 유지
- **XML 파일**: 원본 텍스트 그대로 저장

5.3. DB 처리 기능

5.3.1. 원본 데이터 저장

참조 테이블: `stats_kosis_origin_data`

수집된 통계 데이터를 원본 형태로 저장합니다.

처리 과정:

1. JSON 파일에서 데이터 로드
2. 각 레코드를 데이터베이스 스키마에 맞게 변환
3. 배치 단위로 bulk insert 수행
4. 주요 필드 매핑:
    - `src_data_id` : 통계 소스 데이터 ID
    - `tbl_id` : 통계 테이블 ID
    - `c1` , `c2` , `c3` , `c4` : 분류 코드
    - `c1_nm` , `c2_nm` , `c3_nm` , `c4_nm` : 분류명
    - `itm_id` , `itm_nm` : 항목 ID 및 명
    - `prd_de` : 기준일자
    - `dt` : 데이터 값
    - `stat_latest_chn_dt` : 통계 최신 변경일자
    - `data_ref_dt` : 데이터 참조일자 (오늘)

5.3.2. 통합 테이블 이관

참조 테이블: 통계별 통합 테이블 (예: `stats_dis_hlth_disease_cost_sub` , `stats_dis_reg_natl_by_new` 등)

원본 데이터를 통계별 통합 테이블로 이관합니다.

처리 과정:

1. 기존 데이터 삭제:
    - 동일한 `src_data_id` 및 `src_latest_chn_dt` 를 가진 데이터 중
    - 오늘 생성되지 않은 데이터만 삭제

2. 데이터 변환 및 삽입:
    - `prd_de` : 문자열을 INTEGER로 변환
    - `dt` : 문자열을 NUMERIC으로 변환 (특수 케이스: `-` 또는 빈 문자열은 0으로 변환)
    - 일부 통계는 `dt` 를 문자열로 유지
    - `lst_chn_de` : 빈 문자열은 NULL로 변환 후 DATE 타입으로 변환

특수 처리:

- 통계별로 다른 데이터 타입 변환 규칙 적용
- 통합 테이블 ID는 `stats_src_data_info.intg_tbl_id` 에서 조회

5.3.3. 메타데이터 저장

참조 테이블: `stats_kosis_metadata_code`

통계의 분류체계 및 코드 정보를 저장합니다.

처리 과정:

1. XML 파일 파싱 (선행 비XML 텍스트 제거)
2. `MetaRow` 요소 추출
3. 기존 메타데이터 삭제 (동일 `src_data_id` , `tbl_id` , `stat_latest_chn_dt`)
4. 배치 단위로 bulk insert
5. 주요 필드:
    - `obj_id` , `obj_nm` : 객체 ID 및 명
    - `itm_id` , `itm_nm` : 항목 ID 및 명
    - `up_itm_id` : 상위 항목 ID
    - `unit_id` , `unit_nm` : 단위 ID 및 명

5.3.4. 통계 소스 정보 업데이트

참조 테이블: `stats_src_data_info`

통계 소스의 최신화 정보를 업데이트합니다.

업데이트 항목:

- `stat_latest_chn_dt` : 통계 최신 변경일자 (latest 파일에서 추출)
- `stat_data_ref_dt` : 통계 데이터 참조일자 (오늘)
- `avail_cat_cols` : 사용 가능한 분류 컬럼 목록 (JSON 배열)
    - 실제 데이터에 값이 존재하는 `c1` ~ `c4` 만 포함

- `updated_at` , `updated_by` : 업데이트 시각 및 주체

5.3.5. 시스템 데이터 요약 정보 업데이트

참조 테이블: `sys_data_summary_info`

통합 테이블의 최신화 정보를 업데이트합니다.

업데이트 항목:

- `src_latest_chn_dt` : 소스 최신 변경일자
- `sys_data_ref_dt` : 시스템 데이터 참조일자 (오늘)
- `updated_at` : 업데이트 시각

검증:

- 업데이트 전 레코드 존재 여부 확인
- 업데이트된 행 수 검증 (1개여야 함)

5.3.6. 관리 테이블 업데이트

참조 테이블:

- `sys_stats_src_api_info` : 통계별 API 동기화 정보
- `sys_ext_api_info` : 전체 API 동기화 정보

업데이트 항목:

- `latest_sync_time` : 최신 동기화 시각 (현재 시각)
- `updated_at` , `updated_by` : 업데이트 시각 및 주체

처리 시점:

- `sys_stats_src_api_info` : 각 통계 처리 완료 시
- `sys_ext_api_info` : 모든 통계 처리 완료 후 (한 번만)

5.4. 데이터 정리 기능

5.4.1. 과거 데이터 삭제

모든 데이터 처리가 완료된 후, 최신 데이터만 유지하고 과거 데이터를 삭제합니다.

삭제 대상:

- `stats_kosis_origin_data` : 최신 `stat_latest_chn_dt` 가 아니거나 오늘 생성되지 않은 데이터
- `stats_kosis_metadata_code` : 최신 `stat_latest_chn_dt` 가 아니거나 오늘 생성되지 않은 메타데이터
- 통합 테이블: 최신 `src_latest_chn_dt` 가 아니거나 오늘 생성되지 않은 데이터

삭제 조건:

```
(stat_latest_chn_dt != 최신값) OR
(stat_latest_chn_dt == 최신값 AND DATE(created_at) != 오늘)
```

이 조건으로 인해:

- 다른 날짜의 최신 데이터는 유지
- 오늘 생성된 최신 데이터는 유지
- 과거에 생성된 동일 날짜 데이터는 삭제

5.5. 에러 처리 및 트랜잭션 관리

5.5.1. 트랜잭션 관리

원칙:

- 각 통계별로 독립적인 트랜잭션 사용
- 하나의 통계 처리 실패 시 해당 통계만 롤백
- 다른 통계는 정상적으로 커밋

구현:

- 각 통계마다 별도의 데이터베이스 세션 생성
- ThreadPoolExecutor 내에서 세션 관리
- 예외 발생 시 `session.rollback()` 호출
- 정상 완료 시 `session.commit()` 호출

5.5.2. 에러 처리

API 호출 에러:

- HTTP 상태 코드 200이 아닌 경우 프로그램 종료
- 네트워크 에러 발생 시 프로그램 종료

- Error 31 발생 시 자동 분할 수집 시도

데이터베이스 에러:

- 필수 테이블 누락 시 프로그램 종료
- 데이터 삽입 실패 시 해당 통계 롤백 및 프로그램 종료
- 데이터 타입 변환 실패 시 롤백 및 프로그램 종료

파일 처리 에러:

- 파일 읽기 실패 시 프로그램 종료
- 파일 저장 실패 시 프로그램 종료

5.5.3. 로깅

로그 레벨:

- DEBUG: 상세 디버깅 정보
- INFO: 일반 정보 (기본)
- WARNING: 경고 메시지
- ERROR: 에러 메시지

로그 파일:

- 일반 로그: `logs/YYYYMMDD.log`
- DB 전용 로그: `logs/db_YYYYMMDD.log`

로그 형식:

```
%(asctime)s [%(levelname)s] [PID:%(process)d][%(threadName)s] %(filename)s:%(lineno)d %(funcName
```

부록

부록 A. 주요 설정 항목 상세

A.1. 환경 변수 목록

변수명	설명	기본값	필수 여부
DB_URL	PostgreSQL 연결 문자열	-	필수
LOG_LEVEL	로그 레벨 (DEBUG/INFO/WARNING/ERROR)	INFO	선택
DB_BATCH_SIZE	DB 배치 삽입 크기	100	선택
PARALLEL_WORKERS_FILE	파일 저장 병렬 워커 수	4	선택
PARALLEL_WORKERS_DB	DB 처리 병렬 워커 수	2	선택
EXT_API_INFO_KOSIS_SYS	KOSIS 시스템 식별자	KOSIS	선택
DATA_COLLECTION_SCOPE	데이터 수집 범위 (ALL/PART)	ALL	선택
TARGET_SRC_TBL_ID_LIST	PART 모드 시 대상 통계 목록	-	조건부

A.2. 설정값 권장사항

소규모 환경 (통계 수 < 10개)

- PARALLEL_WORKERS_FILE: 2~4
- PARALLEL_WORKERS_DB: 1~2
- DB_BATCH_SIZE: 100

중규모 환경 (통계 수 10~50개)

- PARALLEL_WORKERS_FILE: 4~6
- PARALLEL_WORKERS_DB: 2~3
- DB_BATCH_SIZE: 200~300

대규모 환경 (통계 수 > 50개)

- PARALLEL_WORKERS_FILE: 6~8
- PARALLEL_WORKERS_DB: 3~4
- DB_BATCH_SIZE: 300~500

부록 B. 로그 파일 구조

B.1. 로그 파일 위치

- 일반 로그: `logs/YYYYMMDD.log`
- DB 전용 로그: `logs/db_YYYYMMDD.log`

B.2. 로그 레벨별 내용

DEBUG 레벨:

- API 응답 내용 (일부)
- 파일 저장 경로 상세 정보
- 설정값 로드 정보

INFO 레벨:

- 프로세스 시작/종료
- 각 단계별 진행 상황
- 처리된 레코드 수
- 성공/실패 메시지

WARNING 레벨:

- 데이터 분할 수집 시도
- 선택적 처리 건너뛰기
- 비정상적이지만 계속 진행 가능한 상황

ERROR 레벨:

- 예외 발생 및 스택 트레이스
- 프로그램 종료 사유

부록 C. 데이터베이스 테이블 및 처리 요약

C.1. 데이터베이스 테이블 참조 요약

조회 전용 테이블:

- `sys_ext_api_info` : 외부 API 정보
- `sys_stats_src_api_info` : 통계 소스 API 정보
- `stats_src_data_info` : 통계 소스 데이터 정보

데이터 저장 테이블:

- `stats_kosis_origin_data` : 원본 데이터
- `stats_kosis_metadata_code` : 메타데이터
- 통계별 통합 테이블: 통합 데이터

정보 업데이트 테이블:

- `stats_src_data_info` : 통계 소스 정보
- `sys_data_summary_info` : 시스템 데이터 요약
- `sys_stats_src_api_info` : 통계별 API 동기화 정보
- `sys_ext_api_info` : 전체 API 동기화 정보

C.2. 주요 처리 단계 요약

1. **초기화**: 환경 검증, API 정보 조회, 디렉토리 생성
2. **데이터 수집**: KOSIS API 호출, 파일 저장
3. **DB 처리** (db 모드):
    - 원본 데이터 삽입
    - 통합 테이블 이관
    - 메타데이터 저장
    - 정보 업데이트
4. **데이터 정리**: 과거 데이터 삭제
5. **동기화 정보 업데이트**: 최신화 시각 갱신