

data_collect 디렉토리 구조

```
| DataCollectApplication.java: 애플리케이션의 실행을 관리
|
|—common
|   FieldKey.java: 네트워크 인터페이스 및 IP 주소와 관련된 다양한 필드를 정의하는 enum
|
|—configuration
|   MessageConsumer.java: 메시지를 소비하며, 수신한 메시지를 처리
|   RabbitMqConfig.java: Rabbitmq의 설정 관리 클래스
|
|—fetcher
|   |—db
|   |   Log.java: DB 테이블과 매핑되는 Log객체에 대한 엔티티 정의,
|   |   LogRepository.java: Log의 엔티티를 관리하기 위한 CRUD 쿼리 메서드 정의
|   |
|   |—service
|   |   LogParserService.java: XML형식의 로그 메시지를 파싱하여 Log객체로 변환하는 기능 제공
|   |   LogProcessingService.java: 파싱된 로그 데이터를 DB에 저장
|   |
|—interfaceIndex
|   |—db
|   |   InterfaceIndexEntity.java: DB 테이블과 매핑되는 InterfaceIndex객체에 대한 엔티티 정의
|   |   InterfaceIndexRepository.java: InterfaceIndex의 엔티티를 관리하기 위한 CRUD 쿼리 메서드
|   |
|   |—service
|   |   InterfaceIndexProcessingService.java: 인터페이스 인덱스를 처리하는 서비스 계층
|   |   InterfaceIndexService.java: 인터페이스 인덱스와 인터페이스 설명을 DB에 저장
|   |
|—mapping
|   |—db
|   |   MappingEntity.java: DB 테이블과 매핑되는 MappingEntity객체에 대한 엔티티 정의
|   |   MappingEntityId.java: MappingEntity객체의 복합 키 정의
|   |   MappingRepository.java: MappingEntity를 관리하기 위한 CRUD 쿼리 메서드
|   |
|   |—service
|   |   MappingMessageService.java: 메시지를 처리하고 DB에 저장하는 서비스 계층
|   |   MappingService.java: MappingEntity를 DB에 저장하고 {TIMESTAMP}.txt파일에 lp 저장
|   |
|—status
|   |—db
|   |   StatusEntity.java: DB 테이블과 매핑되는 StatusEntity객체에 대한 엔티티 정의
|   |   StatusRepository.java: StatusEntity를 관리하기 위한 CRUD 쿼리 메서드
|   |
|   |—service
|   |   StatusService.java: {TIMESTAMP}.txt파일을 바탕으로 StatusRepository 갱신
```

```
└─traffic
  └─db
    | TrafficEntity.java: DB 테이블과 매핑되는 TrafficEntity객체에 대한 엔티티 정의
    | TrafficRepository.java: TrafficEntity를 관리하기 위한 CRUD 쿼리 메서드
    |
    └─service
      TrafficMessageService.java: 메시지를 처리하고 DB에 저장하는 서비스 계층
      TrafficProcessingService.java: TrafficEntity를 DB에 저장
```