LG CNS

DevOn Boot

INTRODUCTION

LG CNS

DevOn Boot 특장점

DevOn Boot는 Container향 상용 프레임워크로 국내 최초 출시되었으며, 금융권의 계정계, 채널계 등 최다 프로젝트 수행 경험을 통해 검증된 기능과 운영 노하우가 반영된 프레임워크 입니다.

지향점		특장점
Spring Boot 기반의 검증된 프레임워크	생산성 / 안정성	▪ 글로벌 표준 기반의 개발 생산성과 운영 안정성 확보
Cloud / Container 최적화	확장성 / 유연성	▪ Cloud / Container 환경에 적합한 경량 구조와 탄력적 확장 지원 ▪ Cloud 환경의 다양한 솔루션 및 MSA 간의 연계 기능 제공 ▪ SAGA Pattern, 동기 보상 등 분산 트랜잭션 기능 제공
금융권 경험을 바탕으로 확장 및 고도화	엔터프라이즈 기능	▪ DevOn Enterprise의 금융권 특화 기능 이식 ▪ 금융권 프로젝트 경험을 통한 기능 개발 ▪ 고성능 처리와 안정적 운영을 위한 기능 제공 ▪ Hot Swap, 분산 캐시 등 무 중단 운영 지원
운영 편의 기능 제공	운영관리 / 모니터링	▪ 도그의 동합 관리 및 Viewer 제공 ▪ 운영 관리를 통한 온라인, 배치 Property 관리 및 실시간 반영 ▪ 내장 WAS의 상태 및 주입된 설정 값 모니터링 기능 제공
지속 가능한 유지보수 지원 체계	유지보수 / 기술지원	▪ 정기 점검 및 유지 보수 서비스 제공 ▪ 정기적인 신규 버전 릴리즈 기반으로 패치 및 기술 지원 제공 ▪ 오픈소스 취약점 조치를 위한 기술 지원 서비스 제공

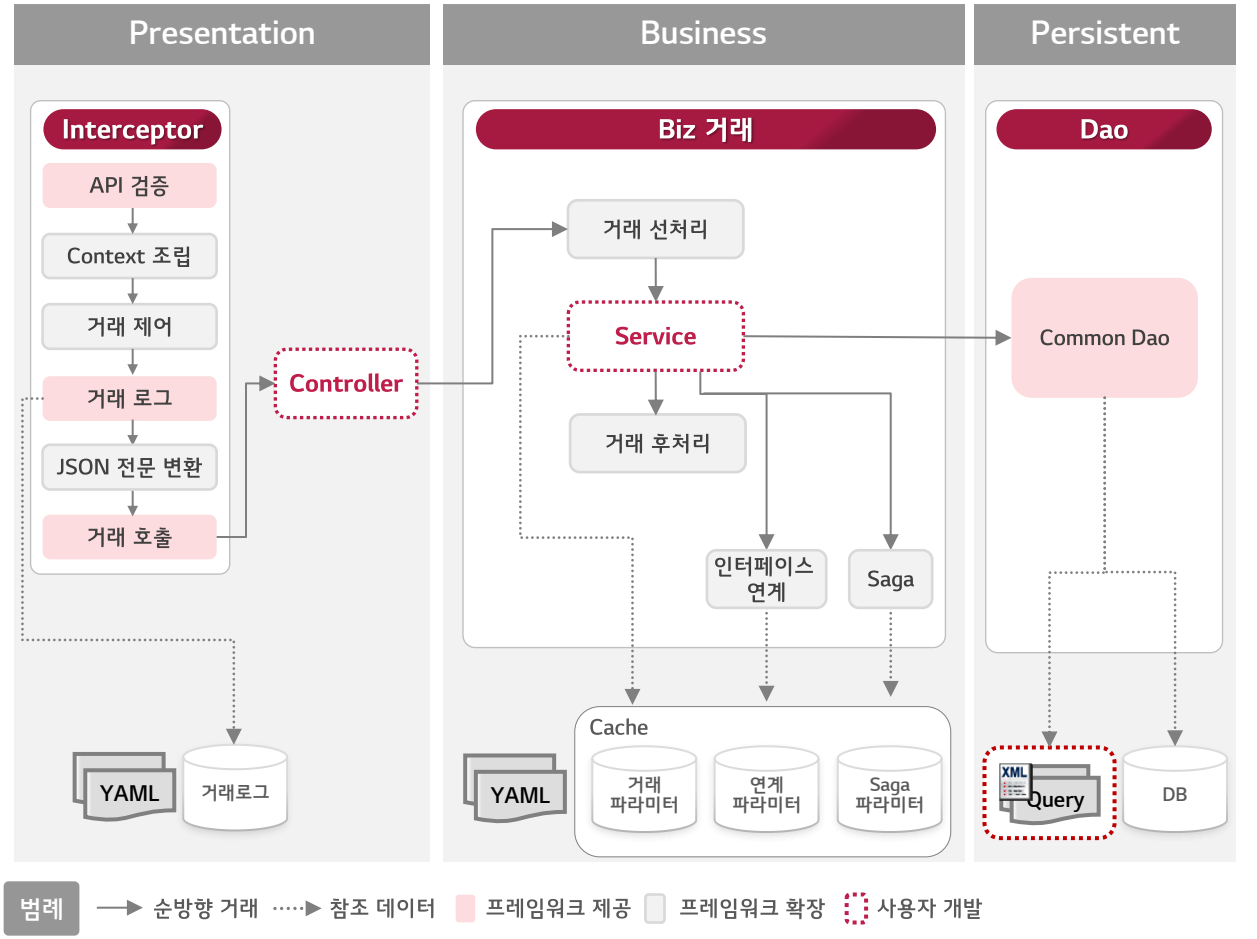
제품 구성도


DevOn Studio
Inspire View
Quick Link
DevOn Perspective
Wizard
Class Reproduction
Report
Outline
AIND
AI-Trans
AI-Analysis
AI-Coding
AI-Test
CCD 개발도구
Reverse
Simple Designer
CoQ
Tester
DevOn Boot EE
MSA
Event Driven
이벤트메시지모니터링
Saga Pattern
Saga처리상태모니터링
배치
일반/온디맨드배치
작업종료/오류관리
실시간모니터링
파일IO구조관리
온라인모듈호출
스케줄러연계
데몬/후행
일반/사용자 데몬
실행/주기 제어
후행 업무로그 적재
업무로그 분배와 처리
후행 재처리
센터컷
순차방식처리
병렬방식처리
혼합방식처리
유량제어
Commit건수 제어
이중기동 방지
운영관리
거래관리
배치작업관리
거래모니터링
Saga모니터링
데몬관리
센터컷관리
후행관리
온라인
분산 캐시
Refresh
거래추적ID
거래로그
거래타임아웃
오류전문생성
전문처리
메시지변환
선/후처리
거래제어
추적로그
연계 처리
Core Engine
압축
Mail
Rest
Validation
XSS
채번
Excel
중복전송방지
조회건수제한
타임아웃
CommonDao
DataAccess (Mybatis)
QueryLoader (File/DB)
Lazyload (query, message)
시뮬레이션거래
대용량처리
인덱스 페이징
스크롤 페이징
추적로그
DB 메시지
시뮬레이션거래
Template Engine (Thymeleaf)
Auto Config
Policy 기반 파일 업로드
파일 다운로드
Composite Transaction 처리
Multi DataSource 처리
Spring
로깅
파일 메시지
Exception
DataSource
Transaction
Starters
IoC
DI
AOP
Web MVC
Test(Junit)
Embedded WAS
Embedded LENA
통합 운영관리
Session Cluster
Config Server
리소스 모니터링
이벤트 수집
이벤트 관리

온라인 아키텍처

DevOn Boot의 온라인 아키텍처는 재사용성을 높이고 업무 구현에만 집중 할 수 있는 아키텍처를 제공하여,
업무 환경 변화에 기민하게 대응하고 제어할 수 있는 구조를 제공합니다.

온라인 실행 흐름도



설명

공통

- Spring Boot 표준 온라인 처리 지원
- 파라미터 기반의 온라인 거래 처리 기능 제공
- Auto-Configuration을 제공하여 간단한 YAML 설정만으로 프레임워크 기능 제공

Presentation

- JSON형태의 전문 데이터에 대해 Business에서 사용하는 VO 객체로 변환
- 데이터 Parsing/검증, 거래제어, 서비스 호출 등의 기능을 수행

Business

- Business에서는 업무 로직만 처리, 거래 파라미터를 통한 거래의 흐름 제어와 선후처리 기능 제공
- Cloud Native 환경에서 마이크로서비스 간, 타 시스템, 레거시 시스템과의 연계 지원 인터페이스 제공

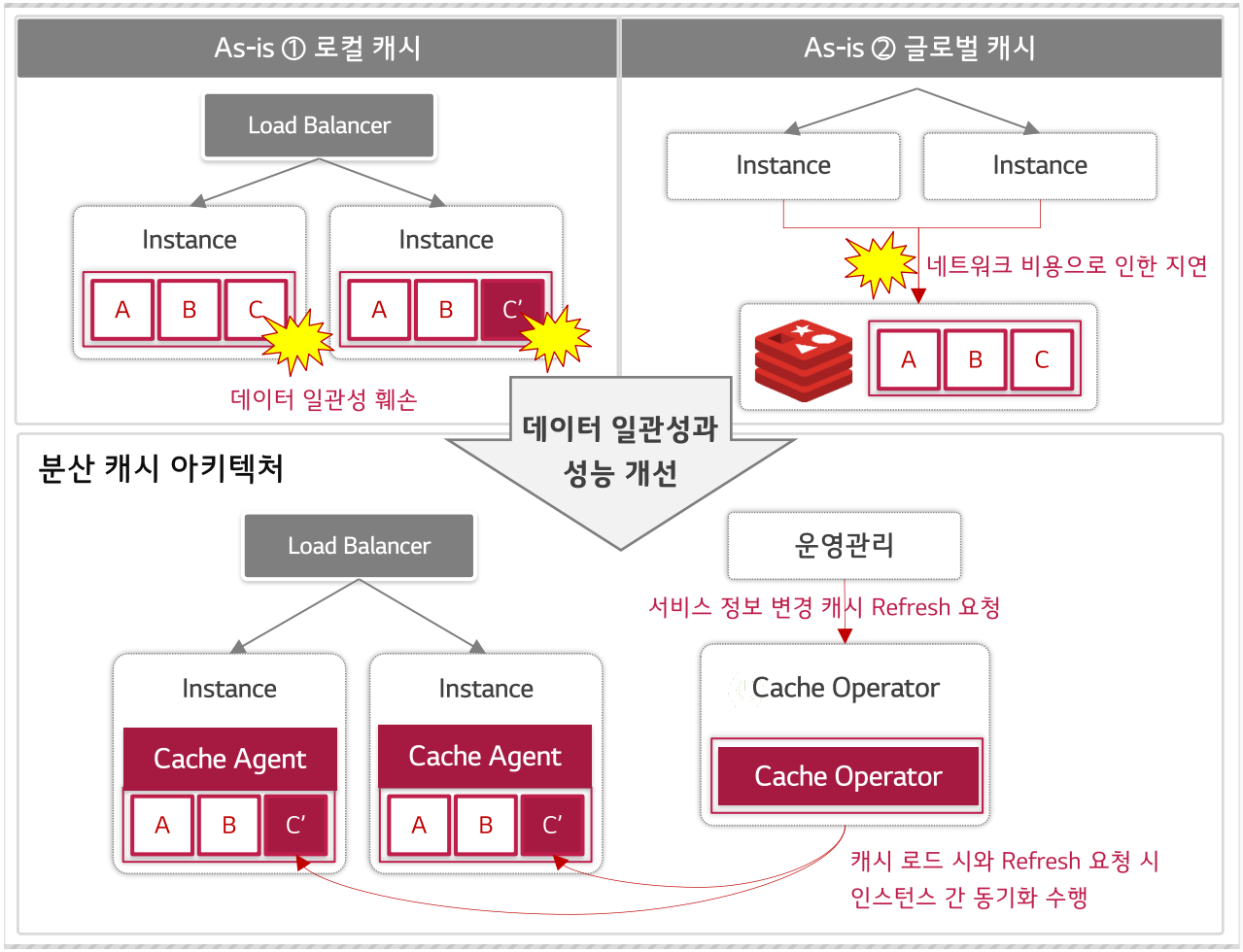
Persistent

- DB Access 처리를 위한 공통 컴포넌트 제공

분산 캐시 동기화/Refresh

자체 캐시 아키텍처를 이용하여 빠르고 일관성 있는 분산 캐시 기능을 제공합니다. Redis와 같은 글로벌 캐시 또한 필요 시 손쉽게 적용 가능합니다.

As-is와 분산 캐시 아키텍처



설명

로컬 캐시

- 각 인스턴스/컨테이너가 각자 자신의 메모리 공간에 캐시를 저장하고 활용
- 속도가 빠르다는 장점 보유
- 분산 환경에서 인스턴스 간 캐시 데이터가 같다는 보장을 할 수 없어 데이터 일관성에 취약

글로벌 캐시

- Redis와 같은 NoSQL 저장소를 중앙에 운용하는 방식
- 분산 환경에서 데이터 일관성 확보 가능
- 네트워크 비용으로 인해 latency가 발생한다는 단점 존재

분산 캐시

- 인스턴스가 각자 자신의 메모리에 캐시를 저장하되, 중앙에서 인스턴스간 데이터를 동기화하는 방식
- 로컬 캐시와 동일한 빠른 Read 성능
- 분산 환경에서 인스턴스간 Cache 데이터를 일관성 있게 유지 가능

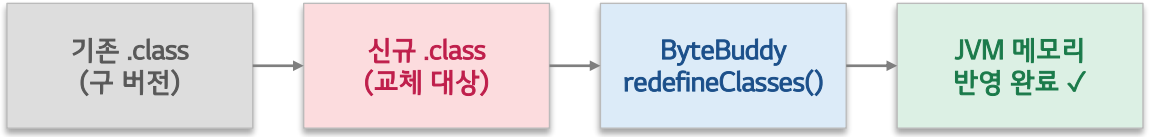
Hot Swap 배포

서비스 중단 없이 운영중인 JVM에 단위 클래스(.class)를 직접 교체하는 무중단 배포 기능을 제공합니다.

Hot Swap 동작 원리 및 제약 사항

- 배포 흐름

- Hot Swap 내부 프로세스 (JVM Instrumentation API)



- 전제조건 및 제약사항
  - WAS의 Exploded WAR 배포 형태 지원 필요
  - 메소드 바디(로직) 변경 한정 지원 : 메소드·필드·클래스 추가/삭제/명칭 변경, 새 클래스 추가 불가
  - AOP 어노테이션 추가·변경 불가 : @Transactional, @Cacheable 등 기동 시점 생성 Proxy 객체는 재생성 불가
  - URL 매핑 추가·변경 불가 : @GetMapping 등 신규 URL 추가 불가. 기존 URL의 핸들러 로직 변경만 가능
  - Hot Swap 적용 시점 : API 호출 이후 유입 요청부터 신규 로직 적용. 이미 실행 중인 트랜잭션은 구 버전 로직으로 처리

주요 특장점

무중단 배포
- 서비스 재시작 없이 운영 JVM에 직접 바이트코드 교체
- 24/265 운영 환경 요건 충족

기존 인프라 활용
- 표준 Oracle/OpenJDK를 그대로 사용
- JVM 교체/debug 개발도구 전환 불필요

CI/CD 파이프라인 연동
- REST API 호출을 통한 런타임 시점 배포
- Shell Script 연동으로 국내/외 대다수상용 배포 툴과 연동 가능

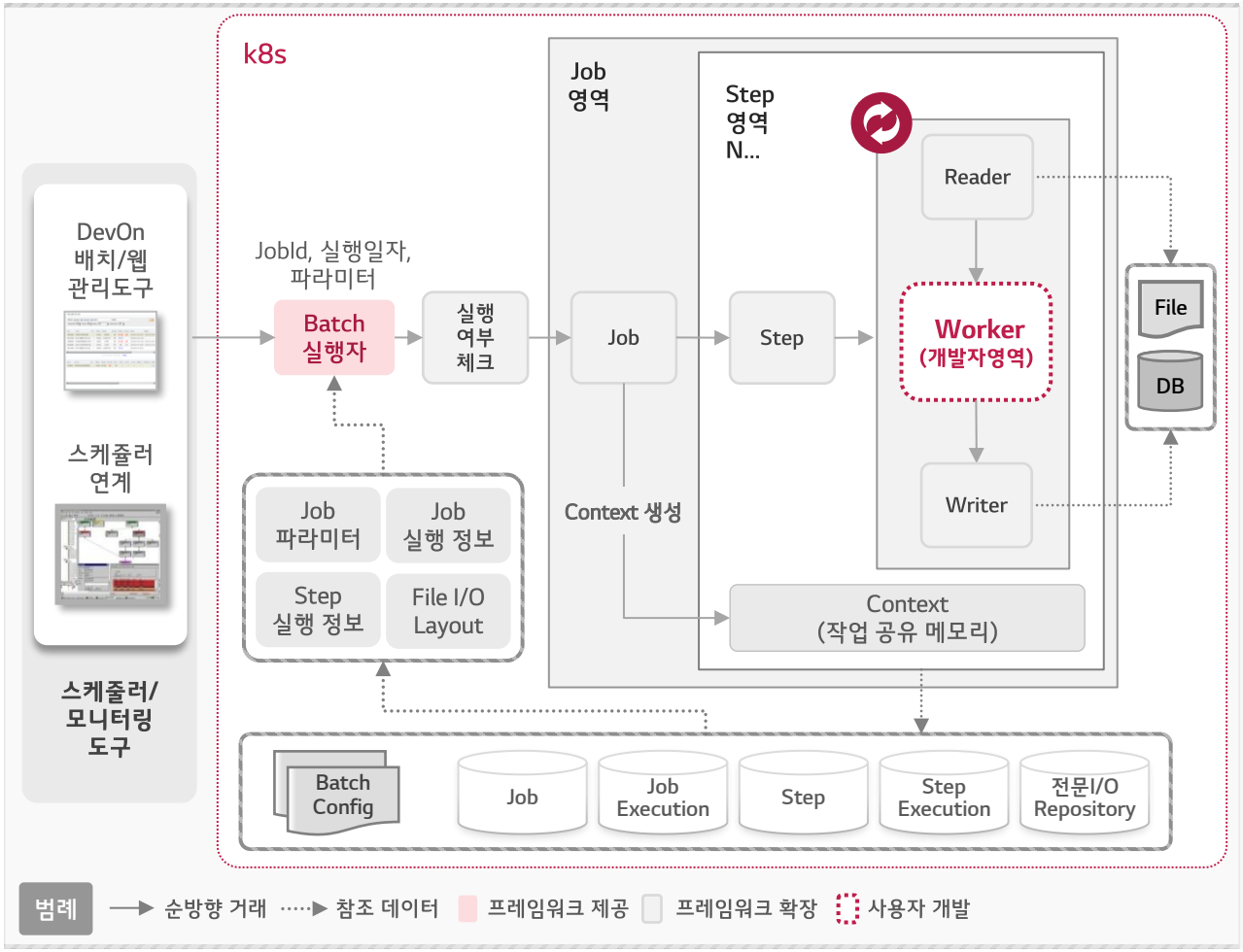
추가 라이선스 비용 無
- DevOn Boot 프레임워크 내장 기능으로 제공

금융권 레퍼런스
- M사 은행권 도입 등 국내 금융권 실 사례 레퍼런스 보유

배치 아키텍처

배치 아키텍처는 배치 업무 개발부터 작업 실행 및 모니터링까지 전반적인 기능을 제공합니다. 다양한 형식의 File I/O를 지원하는 Reader, Writer가 제공되며 웹 관리도구를 이용하여 File I/O 구조를 설계할 수 있습니다.

배치 아키텍처 구성도



주요 기능

다양한 유형의 금융 특화 기능

- DB to DB 배치, File to File(SAM 파일, 일반 파일) 배치, On-demand 배치, 상주(데몬) 배치 기능 제공
- File 및 DB 유형에 맞게 다양한 Reader 및 Writer 제공에 따라 개발 생산성 향상

운영 도구 제공

- File(SAM파일, 일반파일) 구조를 웹 관리도구(IO 구조 정의)로 관리
- 별도의 코드 변경 없이 설정을 통한 IO 구조 정의 기능 제공
- 배치 작업 처리 결과 및 이력을 운영관리 도구로 확인 기능 제공

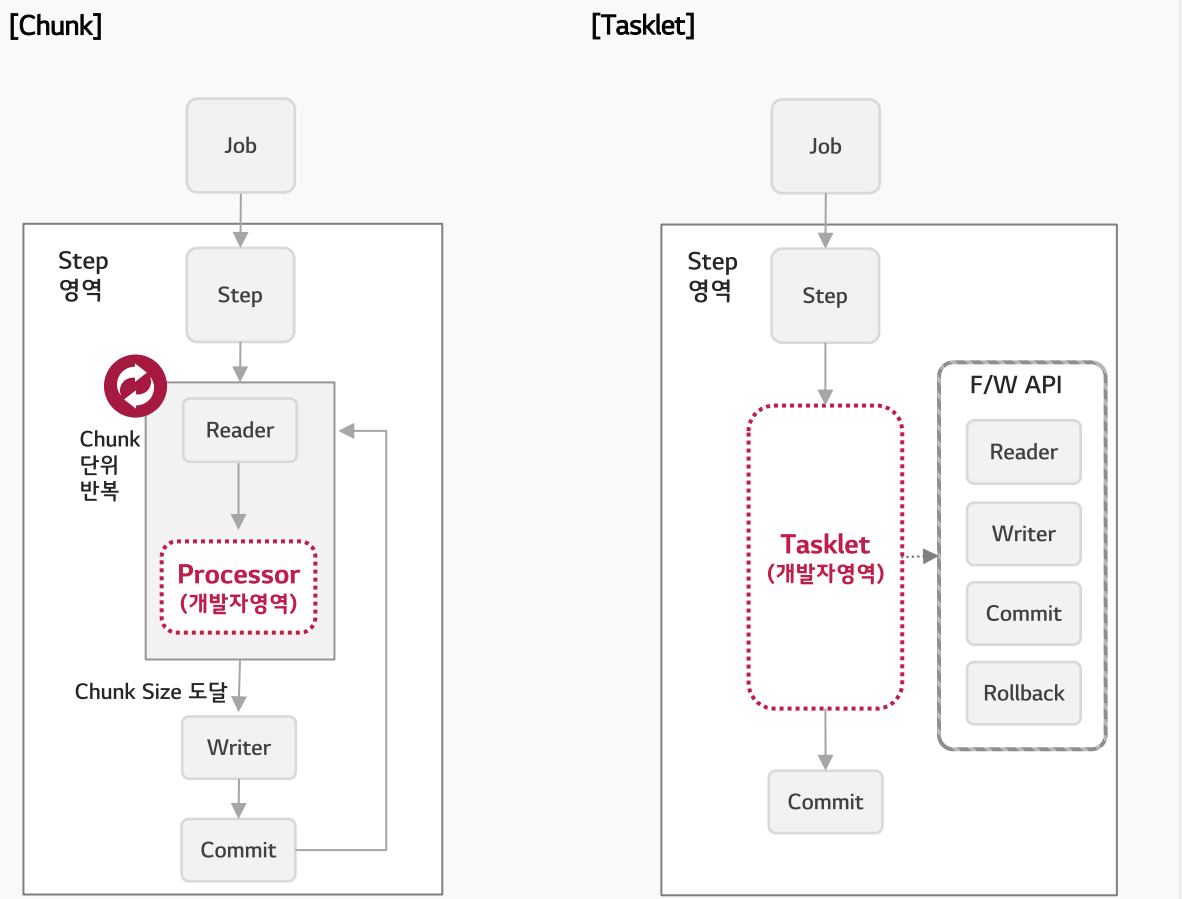
스케줄러 연계

- 스케줄러 연동 가능

배치 처리 방식

Spring의 표준 배치 처리 방식인 Chunk와 Tasklet을 지원하며, 복잡한 금융 배치 처리를 위한 추가 기능들을 제공합니다.

배치 아키텍처 구성도



주요 기능

Chunk

- Spring Batch에서 가장 일반적으로 사용되는 방식
- Reader 에서 읽은 건수만큼 로직이 반복 처리
- 처리 건수가 Chunk Size에 도달할 때마다 Commit 수행
- Chunk Size는 파라미터 기반으로 관리 가능

Tasklet

- Tasklet이 한번만 호출되는 구조
- Taklset내에서 접근할 수 있는 Reader, Writer 제공
- Tasklet내에서 Loop를 통한 반복 처리 작업에서 사용할 수 있는 Transaction 제어 기능 제공
- 오류 발생시 기존 Commit 이후 부터 재처리가 필요한 경우 사용할 수 있는 유틸 및 기능

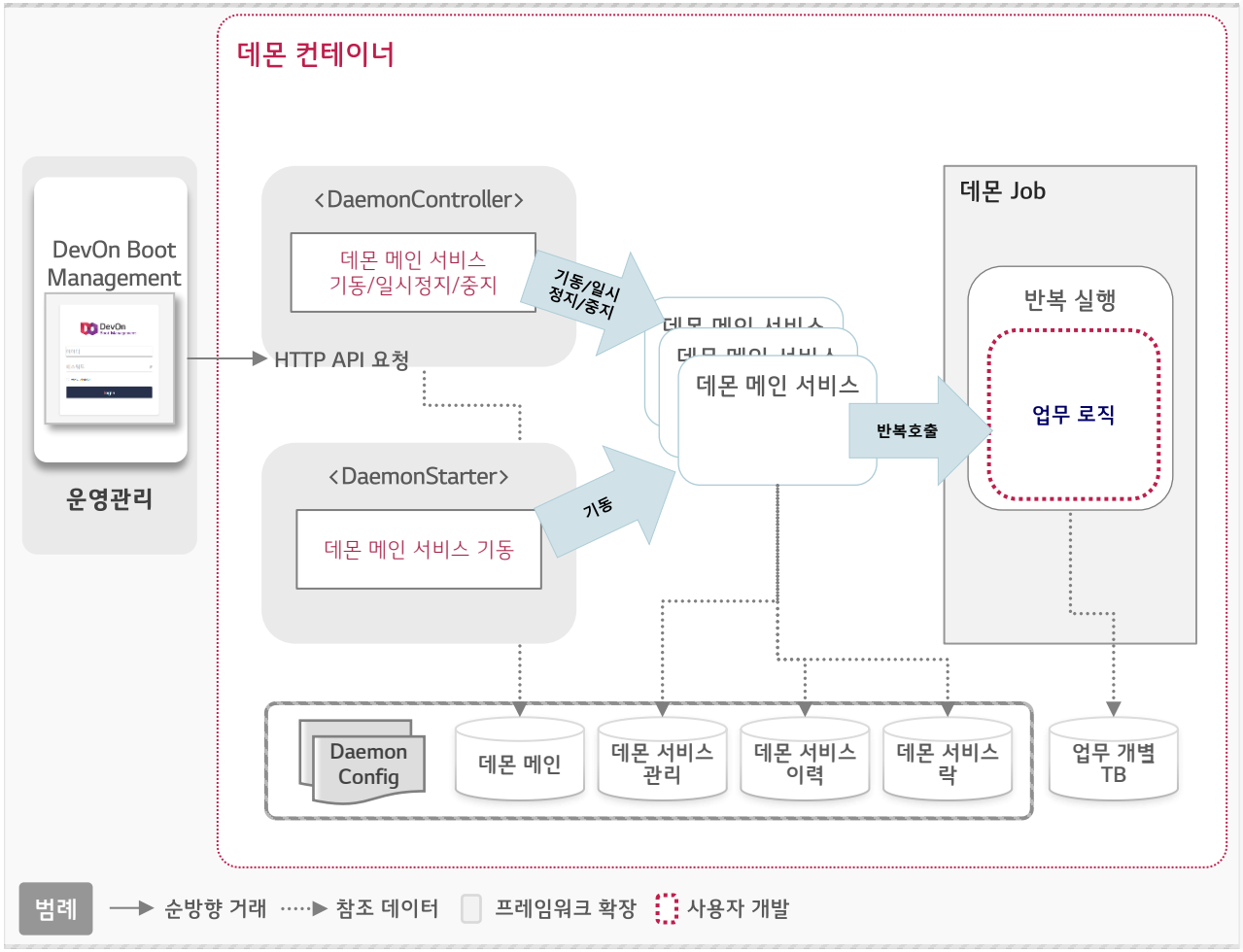
Reader, Writer

- 다양한 유형의 파일 Reader, Writer 제공
- 운영관리 DB파라미터 설정 기반의 Reader, Writer 관리

데몬 아키텍처

데몬 아키텍처는 일정 주기에 따라 반복처리가 필요한 간단한 데몬성 작업에 대한 기능을 지원합니다. 운영관리를 통해 데몬 파라미터를 관리하고 서비스 상태와 이력 모니터링 기능을 제공합니다.

데몬 아키텍처 구성도



주요 기능

주기적으로 반복 실행이 가능한 데몬 기능 제공

- 운영관리를 통한 데몬 서비스의 기동/정지
- 데몬 서비스 자동 시작 기능
- DB fetch 기능 및 트랜잭션 관리 기능을 지원하는 Iterable Daemon 기능 제공
- 사용자 정의 Daemon 기능 제공

운영관리 도구와의 연계

- 운영관리 화면은 통한 데몬 실행/중지 기능
- 파라미터 관리
- 데몬 서비스 상태 및 이력 모니터링 기능

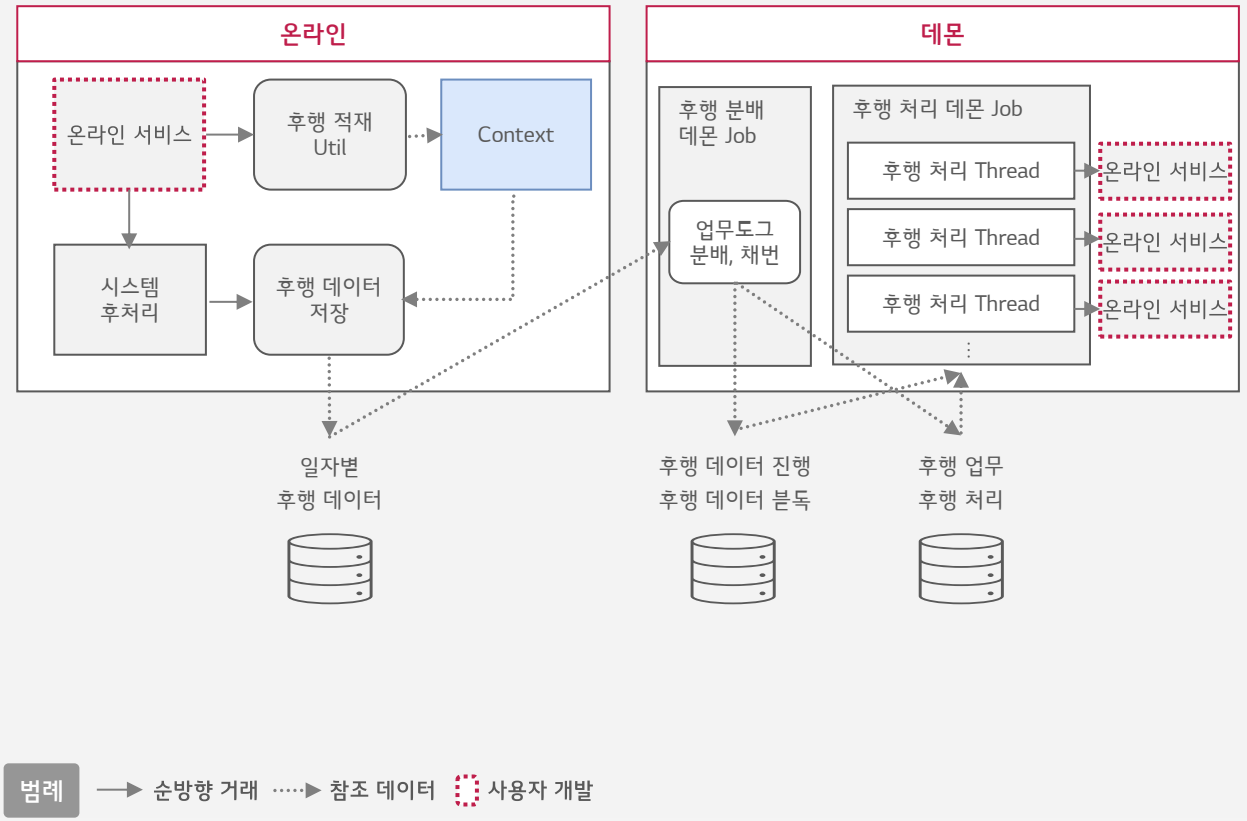
상주 배치 대체

- 가볍고 반복 수행되는 상주 배치성 작업을 데몬을 통해 수행 가능
- Spring Batch의 무거운 프로세스 기동이 불필요

후행 아키텍처

후행은 온라인 서비스에 영향을 최소화 하기 위해 업무로그와 데몬Job을 이용해 후처리 업무를 비동기적으로 실행 할 수 있는 아키텍처를 제공합니다.

후행 아키텍처 구성도



주요 기능

후행 업무로그 적재/저장

- 일자별 업무 로그(후행 데이터) 테이블 제공
- 업무 로그 적재 Util
- 온라인 서비스 종료시점에 후행 데이터 저장

업무로그 분배, 처리 데몬 Job

- 데이터 저장 부하를 줄이기 위한 데이터 분배
- 독립적인 후행 처리 실행 보장

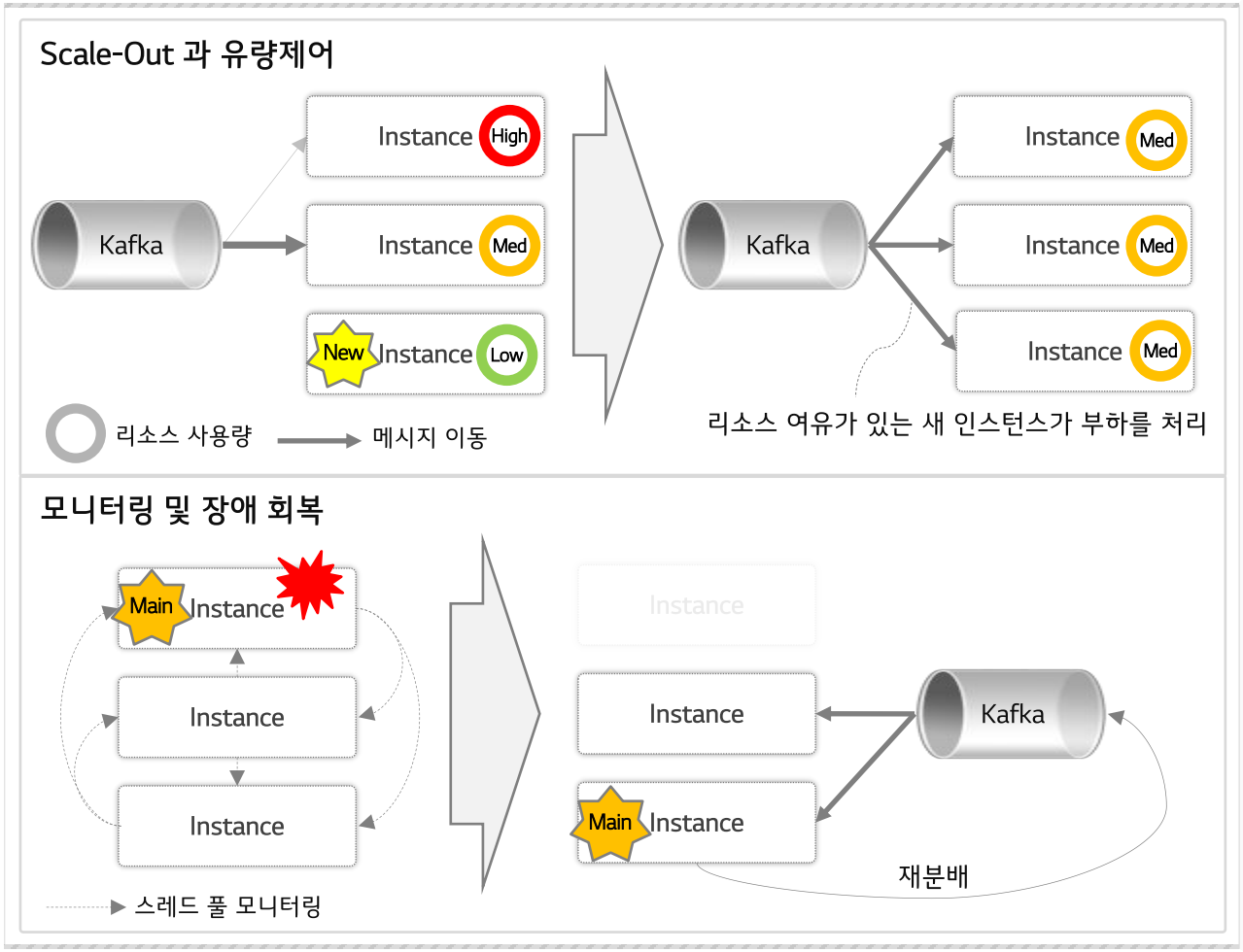
처리 상태 모니터링, 오류 건 재처리

- 데이터별 처리 상태 업데이트
- 오류 건에 대한 주기적 재처리 기능 제공

센터컷 아키텍처

분산 환경의 장점을 활용하여 실시간 Scale-Out을 통한 부하 분산이 가능하고 장애 회복력이 있는 센터컷을 지원합니다.

센터컷 아키텍처



설명

On-Premise 센터컷의 기본 기능 제공

- 온라인 서비스, Filter, Interceptor를 그대로 사용 가능하며 로그 레벨 별도 관리 가능
- 순차/병렬 혼합처리 및 동시성 수준 설정 가능
- 오류 건 재처리, 중지/재개 제어 가능
- 운영관리를 통한 모니터링/집계 제공

Scale-Out과 부하 분산

- 기존의 메시지 push 방식 대신 polling 방식을 채택
- 별도 설정 없이 센터컷 실행 중에 Scale-Out 가능
- 특정 인스턴스의 부하 상황 시 리소스 여유가 있는 인스턴스에 부하가 분산됨

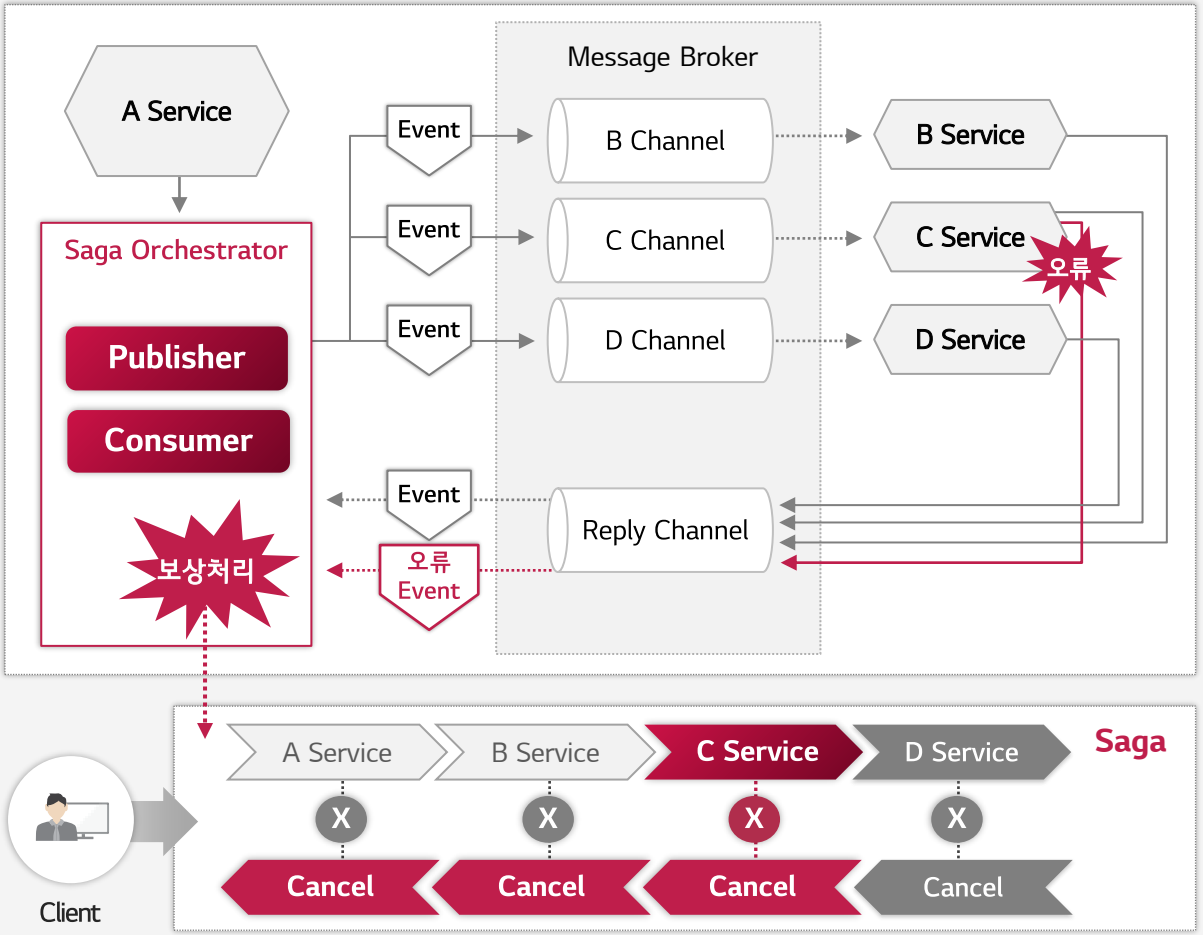
모니터링 및 장애 회복

- 모든 인스턴스가 클러스터로 묶여 서로를 실시간 모니터링, 센터컷 처리 상태의 정합성 보완
- 특정 인스턴스 장애 발생시 메인매니저가 새로 선발되어, 미처리 데이터를 재처리하도록 작동

DevOn Saga

서비스 간 업무 프로세스 파악과 모니터링에 용이한 Orchestration 방식의 Saga 패턴 개발 기능을 제공합니다.

Orchestration Saga 처리 흐름



설명

Orchestration

- 통제자가 각 서비스를 이벤트 기반으로 Request/Reply하여 전체 흐름을 제어함
- 복잡한 Saga에 적합
- 보상 트랜잭션의 최초 시작 서비스에서 생성되는 Saga 인스턴스 제공
- Saga처리 시 History 데이터 및 처리 결과값을 저장
- Scale-in/out에 유연한 아키텍처 제공

이벤트 모니터링

- Saga 처리상태 모니터링
- 각 서비스에서 발행/소비되는 이벤트들에 대한 모니터링 제공

DevOn Boot 운영관리

DevOn Boot Management는 파라미터 관리, 서비스의 등록과 상태 확인, Saga 상태 조회, 메뉴 관리와 사용자 관리 등의 관리자 기능을 제공합니다.

운영 관리 메뉴

DevOn Boot Management

API
- API 관리
- 거래확장파라미터 관리
- API 거래 제어
- 장애거래 제어
- 장애거래 제어 이력

전문
- 전문 관리
- 전문 상세

Saga
- Saga 정보
- 이벤트메시지 정보

로그
- 저널 로그 조회
- 추적 로그 조회
- 연계 로그 조회

컨버터
- 전송 Rule
- 연계 어댑터
- 연계 파라미터
- 루프백 관리
- 타이머 관리
- Mapping 관리

어플리케이션
- 어플리케이션 정보
- 어플리케이션 모니터링

센터컷
- 센터컷 메인
- 센터컷 집계

데몬/후행
- 데몬 메인
- 데몬 서비스 현황
- 데몬 서비스 이력
- 후행 업무 관리
- 후행 처리 관리
- 후행 데이터 조회
- 후행처리 상태조회
- 후행 재처리 등록

배치 관리
- 배치 작업 관리
- 배치 이력 조회
- 배치 사용자정의 집계 조회
- 배치 파라미터 정보 조회
- 배치 파일정보 조회

관리자
- 언어 관리
- 메시지 관리
- 레이블 관리
- 매체 관리
- 기기 관리
- 사용자 관리
- 역할 관리
- 메뉴 관리
- 메뉴역할관리
- 코드 관리
- 카테고리 관리

주요 기능

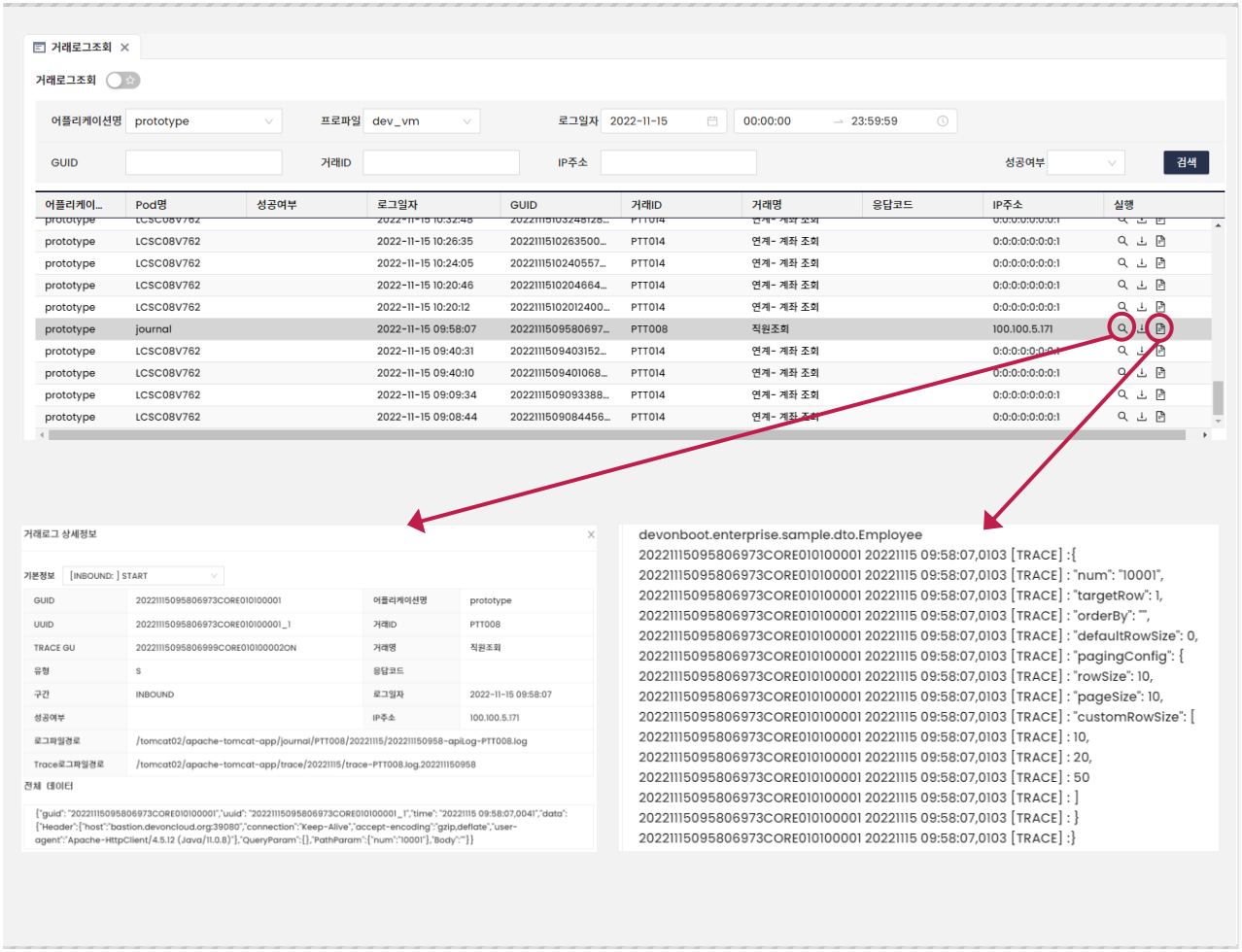
DevOn Boot Management

- Cloud Native 환경에서 거래 파라미터의 관리와 API 등록 및 상태확인 기능 제공
- 타 시스템 또는 레거시 시스템과의 연계를 지원하는 컨버터 파라미터 관리
- 배치 작업 파라미터와 이력 조회 기능 제공
- Saga Instance, Event Message 정보 조회 기능 제공
- 메뉴 관리, 사용자 관리, 권한 관리, 코드 관리 등의 관리자 기능 제공

거래로그 조회

운영관리에서는 DB/File에 기록된 온라인 거래로그의 입출력 데이터를 조회할 수 있는 기능을 제공합니다. GUID, 거래식별자, 성공여부 등의 다양한 조건 기반으로 특정 거래의 로그를 조회할 수 있습니다.

거래로그 조회



주요 기능

다양한 검색 조건

- 거래식별자, 성공여부, GUID 등의 다양한 조건으로 로그 조회

실시간 로그 On/Off

- API 별로 입/출력 거래로그 실시간 조정 (켜기/끄기) 기능

통합 조회 항목 제공

- 거래추적 및 거래상태를 통합 조회할 수 있는 항목 제공 (로그일자, 거래정보, 성공여부, 시스템정보 등)

상세 로그정보 조회

- DB/FILE로 기록된 거래로그의 헤더정보를 포함한 상세 입출력 로그 조회 화면 제공

Trace 로그정보 조회

- 거래 추적을 위한 Trace 기능 제공

DevOn Boot 적용 사례(1/2)

제품 적용 사례

사업명	공급일자	고객사	솔루션 역할
디지털 차세대 플랫폼 구축	2026	I 은행	프레임워크
디지털 차세대 구축	2026	N 보험	프레임워크
스마트 창구 시스템 구축	2026	H 카드	프레임워크
인증서 시스템 구축	2026	H 은행	프레임워크
전자주주총회 시스템 구축	2025	K 기관	프레임워크
차세대 시스템 구축	2025	K 기관	프레임워크
차세대 시스템 구축	2025	M 생명	프레임워크(MSA)
인터넷 뱅킹 재구축	2024	H 은행	프레임워크
스마트금융시스템 구축	2024	E 공제조합	프레임워크
개인뱅킹 구축	2023	S 금고	프레임워크
차세대 시스템 구축	2023	S 증권	프레임워크(MSA)
카드시스템 구축	2023	K 뱅크	프레임워크(MSA)
차세대시스템 구축	2023	S 기관	프레임워크
정보계 재구축	2022	H 은행	프레임워크

DevOn Boot 적용 사례(2/2)

제품 적용 사례

사업명	공급일자	사업개요	솔루션 역할
보험부문 공동구축	2022	K 보험	프레임워크
L* 플랫폼 차세대 구축 사업	2022	L 면세점	프레임워크
차세대 시스템 구축	2022	S 백화점	프레임워크
신용평가시스템 구축	2022	T 신용평가	프레임워크(MSA)
OPEN 통합 영업플랫폼 구축	2022	H 생명	프레임워크(MSA)
The N* 시스템 구축	2021	S 은행	프레임워크(MSA)
판토스 차세대 포워딩 시스템 구축	2021	P 묻듀	프레임워크
마이데이터 플랫폼	2021	K 은행	프레임워크(MSA)
리* Reboot 구축	2021	K 은행	프레임워크(MSA)
전산시스템 재구축 사업	2021	S 건설	프레임워크
모바일 SFA 시스템 구축	2020	H 생명	프레임워크

감사합니다

문의: devon@lgcns.com