# 인터페이스 구현

## 1. 인터페이스 설계 확인

### (1) 외부, 내부 모듈 간 공통 기능 및 데이터 인터페이스 확인

1) **인터페이스 설계서**

▼ 개념

- 이 기종 시스템 및 컴포넌트 간 **데이터 교환 및 처리를 위해** 각 시스템의 교환되는 **데이터, 업무, 송수신 주체** 등이 정의된 문서
- 인터페이스 현황 파악 목적으로 **인터페이스 목록** 및 각 인터페이스의 **상세 데이터 명세**, **각 기능의 세부 인터페이스 정보** 정의한 문서

▼ 목록

- **인터페이스 번호** 및 인터페이스가 되는 **시스템의 정보** 및 관련 요구사항 **ID**를 목록 형태로 보여줌
- 인터페이스 번호, 연계 방식, 통신 유형, 처리 유형, 주기, 데이터 형식 등

▼ 명세

- 인터페이스 번호 당 인터페이스가 되는 **데이터**, **데이터 형식, 송수신 시스템의 정보** 구체화 한 문서

▼ 상세 기능 인터페이스 정의서

- 데이터 송수신 시스템 간의 **데이터 저장소**와 **속성** 등의 상세 내역을 포함
- 인터페이스 ID - 인터페이스 구분 위한 식별자. 업무 분류 코드와 연속 번호를 같이 사용
- 사전 조건 - 인터페이스의 세부 동작이 정상적으로 작동하기 위한 **사전에 완료되어야 하는 조건 기술**
- 사후 조건 - 인터페이스의 세부 동작이 **정상적으로 작동된 이후에 발생되는 조건 기술**
- 파라미터 - 인터페이스 **구성 항목 값**
- 반환 값 - 인터페이스 전송 후 **반환되는 값**

**2) 정적, 동적 모형 및 데이터 명세에 따른 인터페이스 설계서**

▼ 정적, 동적 모형을 통한 인터페이스 설계서

- 정적, 동적 모형을 통해 각 시스템의 구성 요소를 표현한 다이어그램 활용하여 시스템, 컴포넌트별 인터페이스와 요구 조건 확인
- 시스템을 구성하는 **주요 구성 요소 간 트랜잭션 확인**을 통해 시스템에서 인터페이스와 인터페이스를 통해 **상호 교환되는 트랜잭션을 확인**

▼ 데이터 명세를 통한 인터페이스 설계서

- 제공하는 인터페이스 서비스에 대한 상세 명세를 표현
- 제공하는 서비스 목록, 인터페이스 방식 및 명세, 리턴 형태까지 정의를 상세하게 표현

### 3) 내부, 외부 모듈 간 공통 기능 및 데이터 인터페이스 확인 방안

i. 내부, 외부 모듈의 기능 확인

▼ 인터페이스 정의서를 통한 기능 확인

- 시스템 인터페이스 정의서
- 상세 기능 인터페이스 정의서

▼ 정적, 동적 모형을 통한 기능 확인

- 인터페이스가 표현된 정적, 동적 다이어그램을 통해 내불, 외부 모듈 기능 확인

ii. 공통적으로 제공되는 기능과 데이터의 인터페이스 확인

▼ 인터페이스 설계서를 통한 공통 기능 확인

- 공통적으로 제공되는 기능 확인

▼ 인터페이스 설계서를 통한 데이터 인터페이스 확인

- 공통 기능을 식별하고, 이를 중심으로 필요한 데이터 인터페이스 항목 확인

## (2) 외부 및 내부 모듈 연계를 위한 인터페이스 기능 식별

### 1) 외부, 내부 모듈 연계 방법

▼ EAI

- 기업에서 운영되는 서로 다른 플랫폼 및 애플리케이션 간의 정보를 전달, 연계, 통합이 가능하도록 해주는 솔루션
- 각 비즈니스 간 통합 및 연계성을 증대시켜 효율성을 높여 주고 각 시스템 간의 확장성을 높임
- 유형

**포인트 투 포인트** - **1:1** 단순 통합 방법. 솔루션을 구매하지 않고 개발자 간의 커뮤니케이션을 통해서도 통합 가능

**허브 앤 스포크** - 단일한 접점의 **허브** 시스템을 통하여 데이터를 전송하는 중앙 집중식 방식. 허브 장애 시 전체 장애 발생

**메시지 버스** - 애플리케이션 사이 미들웨어(버스)를 두어 연계하는 미들웨어 통합 방식. 뛰어난 확장성, 대용량 데이터 처리 가능

**하이브리드 -** 그룹 내 허브앤 스포크, 그룹 간 메시지 버스 방식 사용. 그룹 내 환경에 맞는 작업 가능

▼ 특징

- 수행 목적 - 기업 내부의 이기종 응용 모듈 간 통합
- 토폴로지 - 포인트 투 포인트, 허브 앤 스포크, 메시지 버스, 하이브리드
- 핵심 기술 - 어댑터, 브로커, 메시지 큐
- 통합 형태 - 애플리케이션 간의 단단한 통합
- 적용 영역 - 기업 내부망

▼ ESB

- 기업에서 운영되는 서로 다른 플랫폼 및 애플리케이션들 간을 **하나의 시스템으로 관리 운영**할 수 있도록 **서비스 중심의 통합**을 지향하는 아키텍처
- **버스를** 중심으로 각각 프로토콜이 호환할 수 있도록 애플리케이션의 통합을 느슨한 결합 방식으로 지원.
- 느슨한 결합: 특정 서비스를 변경 하더라도 연결된 다른 서비스에는 영향을 주지 않는 유연한 구조

▼ 특징

- 수행 목적 - 기업 간의 서비스 교환을 위해 표준 API로 통합
- 토폴로지 - ESB의 분산형 토폴로지 구성
- 핵심 기술 - 웹 서비스, 지능형 라우터, 포맷 변환, 개방형 표준
- 통합 형태 - 서비스 간의 느슨한 통합
- 적용 영역 - 기업 외부 채널망

**2) 외부, 내무 모듈 연계를 위한 인터페이스 기능 식별 절차**

- 외부 및 내부 모듈 간 연계된 기능 식별

시나리오로 식별

- 연계된 기능에 따른 인터페이스 기능 식별

**3) 외부 및 내부 모듈 간 인터페이스 데이터 표준 확인**

- 상호 연계하고자 하는 시스템 간 인터페이스가 되어야 할 범위의 데이터 형식과 표준을 정의
- 인터페이스 데이터 전송 시 인터페이스 데이터 형태가 동일한 경우는 그대로 전송, 동일하지 않은 경우 데이터를 변환하여 전송
- 송수신 시스템 간의 인터페이스 데이터를 표준화하기 위해 **송수신 데이터 중 공통의 영역을 추출하여 정의**하는 경우, **한쪽의 데이터를 변환하는 경우**가 있다.

▼ 절차

- 식별된 데이터 인터페이스를 통해 인터페이스 데이터 표준 확인

i. 데이터 인터페이스 입출력 의미 파악

ii. 입출력 의미 파악을 통한 데이터 표준 확인

- 인터페이스 데이터 항목 식별

식별된 인터페이스 기능을 통해 인터페이스 데이터 항목을 식별

필요 데이터 항목과 이전에 식별된 데이터 인터페이스 항목에서 수정, 추가, 삭제 되어야 할 항목이 있는지 검토

- 데이터 표준 최종 확인

식별된 인터페이스 기능 및 데이터 항목을 통해 필요한 데이터 표준 및 조정해야 할 항목을 검토 및 확인

송수신 시스템 간 인터페이스 데이터 표준을 최종적으로 확인

# 2. 인터페이스 기능 구현

## (1) 정의

### 1) 모듈 간 세부 설계서 확인

- 컴포넌트 명세서 - 컴포넌트의 개요, 내부 클래스의 동작, 인터페이스를 통해 외부와 통신하는 명세를 정의
- 인터페이스 명세서 - 컴포넌트 명세서에 명시된 인터페이스 클래스의 세부적인 조건 및 기능을 명시한 명세서

### 2) 일관되고 정형화된 인터페이스 기능 정의

- 분석된 인터페이스의 기능, 데이터 표준, 모듈 설계 명세서를 통하여 일관되고 정형화된 인터페이스 기능 정의

### 3) 정의된 인터페이스 기능에 대한 정형화

- 정의된 인터페이스 기능을 표준화하고 사람들이 보기 쉽게 정형화
- 특정 하드웨어나 소프트웨어에 의존적이지 않게 작성, 가독성을 높이기 위해 **프로세스 형태**나 **유스케이스 다이어그램의** 형태로 작성

## (2) 구현

### 1) 분석

- 상세하게 정의된 기능 구현 정의 내용을 토대로 어떻게 구현할 것인지 분석

### 2) 구현

▼ 데이터 통신을 사용하는 인터페이스 구현

- 인터페이스 객체 생성 구현

인터페이스 객체를 생성하기 위해 데이터베이스에 있는 정보를 SQL을 통하여 선택한 후 이를 JSON으로 생성

- 인터페이스 객체 전송 후 전송 결과를 수신 측에서 반환 받도록 구현

송신 측에서 JSON으로 작성된 인터페이스 객체를 AJAX 기술을 이용하여 수신 측에 송신

수신 측에서는 제이슨 인터페이스 객체를 수신 받고 이를 파싱 후 처리

수신 측의 처리 결과 값은 송신 측에 True, False 값을 전달하여 인터페이스 성공 여부를 확인 전달

▼ 인터페이스 개체를 사용하는 인터페이스 구현

- 송신 시스템의 인터페이스 테이블

송신 관련 정보를 관리하기 위한 항목과 송신 시스템에서 필요한 항목 구현

데이터 전송을 위해 DB Connection이 수신 측 인터페이스 테이블과 연계되도록 구현

프로시저, 트리거, 배치 작업 등의 방법을 통해서 수신 테이블로 데이터를 전송하도록 구현

- 수신 시스템의 인터페이스 테이블

수신 관련 정보를 관리하기 위한 항목과 수신 시스템에서 필요한 항목 구현

인터페이스 데이터를 읽은 후 사전에 정의된 데이터 트랜잭션을 진행할 수 있도록 구현

오류 발생 시 오류 코드 칼럼에 정의된 오류 코드와 오류 내용을 입력하도록 구현

## (3) 예외 처리 방안

### 1) 데이터 통신을 사용한 인터페이스에서 예외 처리 방법

- 송신 측에서 예외 처리 방법

AJAX 호출 후 반환 값을 받아 어떻게 처리 할지를 호출하는 부분에서 사전 정의

- 수신 측에서 예외 처리 방법

수신 측에서 받은 JSON 객체를 처리 시에 try ~ catch 구문 이용하여 예외 처리하고 송신 측에 전달

**2) 인터페이스 개체를 사용하는 인터페이스에서 예외 처리 방법**

예외 처리 메시지와 함께 예외 처리가 발생한 원인을 인터페이스 이력에 기록

- 송신 시스템의 인터페이스 테이블

예외 유형에 따른 예외 코드와 상세한 원인을 함께 입력

- 수신 시스템의 인터페이스 테이블

수신 측에서 데이터가 없거나 잘못된 값을 읽을 경우 예외 발생

예외 발생 시 사전에 정의된 예외 코드를 입력하고 예외 발생 사유를 함께 기록

## (4) 보안 기능 적용

**1) 취약점**

- 데이터 통신 시 데이터 탈취 위협

스니핑(데이터만 몰래 들여다보는 수동적 공격 기법)을 통해 데이터 전송 내역을 감청하여 데이터를 탈취

- 데이터 통신 시 데이터 위변조 위협

전송 데이터에 대한 삽입, 삭제, 변조 공격을 통한 시스템 위협

**2) 구현 방안**

▼ 시큐어 코딩 가이드 적용

- 입력 데이터 검증 및 표현

프로그램 입력 값에 대한 검증 누락, 부적절한 검증, 잘못된 형식 지정

→ 입력 데이터에 대한 유효성 검증 체계 수립, 실패 시 처리 기능 설계 및 구현

- 보안 기능

인증, 접근 제어, 기밀성, 암호화, 권한 관리 등의 부적절한 구현

→ 인증, 접근 통제, 권한 관리, 비밀번호 등의 정책이 적절하게 반영되도록 설계 및 구현

- 시간 및 상태

병렬 시스템, 하나 이상의 프로세스가 동작하는 환경에서 시간 및 상태의 부적절한 관리

→ 공유 자원의 접근 직렬화 (데이터 구조나 오브젝트 상태를 다른 컴퓨터 환경으로 저장하고 재구성할 수 있는 포맷으로 변환하는 과정), 병렬 실행 가능 프레임워크 사용, 블록문 내에서만 재귀 함수 호출

- 에러 처리

에러 미처리, 불충분한 처리 등으로 에러 메시지에 중요 정보가 포함

→ 에러 상황을 처리하지 않거나, 불충분하게 처리되어 중요 정보 유출 등 보안 약점 발생하지 않도록 시스템 설계 및 구현

- 코드 오류

코딩 오류

→ 코딩 규칙 도출 후 검증 가능한 스크립트 구성과 경고 순위의 최상향 조정 후 경고 메시지 코드 제거

- 캡슐화

기능성이 불충분한 캡슐화로 인해 인가되지 않은 사용자에게 데이터 누출

→ 디버거 코드 제거와 필수 정보 외의 클래스 내 private 접근자 지정

- API 오용

의도된 사용에 반하는 방법으로 API를 사용, 보안에 취약한 API 사용

→ 개발 언어별 취약 API 확보 및 취약 API 검출 프로그램 사용

▼ 데이터 베이스 보안 적용

▼ 암호화 알고리즘

- 대칭 키 암호화 알고리즘 - 암, 복호화에 같은 암호키 사용
- 비대칭 키 암호화 알고리즘 - 공개키 공유, 비밀키는 키의 소유자만이 알 수 있음.
- 해시 암호화 알고리즘 - 해시 값으로 원래 입력 값을 찾아낼 수 없는 일방향성의 특성을 가짐

▼ 암호화 기법

- API 방식

애플리케이션 레벨에서 암호 모듈을 적용하는 애플리케이션 수정 방식

→ 애플리케이션 서버에 암, 복호화, 정책 관리, 키 관리 등의 부하 발생

- Plug-In 방식

DB 레벨의 확장성 프로시저 기능 이용, DBMS에 Plug-In 모듈로 동작

→ DB 서버에 암, 복호화, 정책 관리, 키 관리 등의 부하 발생

- Hybrid 방식

API 방식과 Plug-In 방식 결합

→ DB 서버와 애플리케이션 서버로 부하 분산

▼ 중요 인터페이스 데이터의 암호화 전송

- 민감한 정보를 통신 채널을 통하여 전송 시 반드시 암, 복호화 과정을 거쳐야 함
- IPSec, SSL/TLS 등 보안 채널 활용하여 전송

**3) 인터페이스 보안 기능 적용 프로세스**

▼ 인터페이스 각 구간의 보안 취약점 분석

- 송수신 영역의 구현 기술 및 특징을 구체적으로 분석
- 송수신 영역의 구현 기술은 보안 취약점이 발생할 수 있는 영역을 구분하여 분석
- 분석된 인터페이스 기능을 대상으로 분석

- 시나리오 가정

▼ 분석된 보안 취약점을 근거로 인터페이스 보안 기능 적용

▼ 네트워크 구간에 대한 보안 기능 적용

데이터 탈취 위변조를 막기 위해 네트워크 트래픽에 대한 암호화를 적용하고 전송 구간 암호화를 구현

- 전송 계층 보안

상대방 인증 적용, 데이터 기밀성 보장 필요

- 응용 계층 보안

서버만 공개키 인증서를 가지고 통신(위험 분산), 연결 단위 외 메시지 단위로도 인증 및 암호화 필요

▼ 애플리케이션 보안 기능 적용

애플리케이션 구현 코드 보안 취약점을 보완하는 방향으로 보안 기능 적용

시큐어 코딩 가이드 참조

- 비인가자 접근 권한 관리
- 악의적 코드 삽입 금지
- 악의적 시도 시 에러 처리

▼ 데이터베이스 보안 기능 적용

데이터베이스의 접근 권한 강화 및 데이터베이스 동작 객체(SQL, 프로시저, 트리거 등)의 보안 취약점 제거를 위해 보안 기능 적용

민감 데이터의 경우 데이터 자체의 보안 방안(암호화, 익명화 등)도 고려

- 데이터베이스 접근 권한
- 악의적 코드 삽입 금지
- 민감 데이터 관리
- 악의적 시도 시 에러 처리

# 3. 인터페이스 구현 검증

## (1) 검증

### 1) 검증 도구

▼ 개념

- 인터페이스 구현 및 감시 도구를 통해서 인터페이스 동작 상태를 검증하고 모니터링
- 인터페이스 세부 기능을 기능 단위로 테스트하는 **단위 테스트**, 전체 인터페이스 흐름을 확인할 수 있는 시나리오를 통한 **통합 테스트** 필요

▼ 종류

xUnit - 함수나 클래스 같은 서로 다른 구성 원소(단위)를 테스트할 수 있게 해줌

Selenium - 테스트 스크립트 언어 학습 필요 없이 기능 테스트 만들기 위한 도구

### 2) 감시 도구

- 애플리케이션 모니터링 툴(APM)을 사용해 동작 상태 감시
- 데이터베이스, 웹 애플리케이션의 트랜잭션과 변수값, 호출 함수, 로그 및 시스템 부하 등 종합적인 정보를 조회, 커넥션 풀 등 지속적인 모니터링이 필요한 자원을 효과적으로 관리

### 3) 검증 프로세스

i. 인터페이스 명세서를 통한 구현 검증에 필요한 요건 분석

- 인터페이스 명세서의 세부 기능을 참조하여 구현 검증 및 감시에 필요한 기능 분석
- 각 기능의 특징에 맞게 구현 검증의 요건 도출

▼ 요건 분석

- 송신 측에서 인터페이스 대상 선택 전송

입력한 대상과 생성된 인터페이스 객체의 정보가 일치하는지 확인

- 인터페이스 객체 전송

암호화된 통신으로 올바르게 수신측에 전달되었는지 확인

전달된 정보가 수신된 정보와 일치하는지 확인

파싱된 정보가 송신된 정보와 일치하는지 확인

- 수신 후 수신 측 트랜잭션과 결과 반환

수신된 데이터와 연관 있는 이후 트랜잭션의 결과 값과 일치 여부

ii. 감시 및 검증 도구 준비

iii. 검증 수행

- 외부 시스템(송수신)과 연계 모듈(송수신)의 동작 상태를 인터페이스 구현 검증 도구를 통해 확인
- 예상되는 결과 값과 검증 값 비교

최초 데이터 입력 → DB에서 조회 → 송신 객체 생성 → 송신 객체 전송 → 수신 및 파싱 → 데이터 트랜잭션 → 수신 결과 반환

- 외부 모듈이 서비스를 제공하는 동안 인터페이스 동작이 정상적으로 수행 되는지 감시 도구를 통해 확인 가능

## (2) 인터페이스 오류 처리 확인 및 보고서 작성

### 1) 인터페이스 오류 처리 방법

i. **사용자 화면에서 오류를 인지하도록 구현**

- 알람 형태로 화면에 표시, 주로 실시간으로 데이터가 인터페이스 되는 경우 사용

ii. **인터페이스 오류 로그 생성**

- 오류 시 관련 오류 로그 생성
- 시스템 관리자나 운영자가 오류 로그 확인 가능

iii. **인터페이스 관련 테이블에 오류 사항 기록**

- 인터페이스 트랜잭션 기록을 별도로 보관하는 경우 테이블에 오류 사항 기록

### 2) 보고서 작성

- 인터페이스 오류 발생 시 상황 인지 및 조치 사항을 시간 경과에 따라 작성
- 최초 발생 보고, 오류 처리 경과 보고, 완료 보고