

# 인터페이스 구현

## chap1. 인터페이스 설계 확인

- 외부, 내부 모듈 간 공통 기능 및 데이터 인터페이스 확인
  - 인터페이스 산출물
    - 사용자 인터페이스와 시스템 인터페이스로 구분
    - 소프트웨어 개발 단계별 인터페이스 산출물
      - 분석 단계
      - 명세 단계
      - 설계 단계
  - 인터페이스 설계서
    - 이 기종 시스템 및 컴포넌트 간 데이터 교환 및 처리를 위해 각 시스템의 교환되는 데이터, 업무, 송수신 주체 등이 정의된 문서.
    - 시스템 인터페이스 목록 사례
      - 인터페이스 번호, 일련번호, 인터페이스id, 인터페이스 명, 송신 시스템, 수신 시스템, 대내외 구분, ..
    - 시스템 인터페이스 정의서
    - 상세 기능별 인터페이스 명세서 (**아명오개 전후파반**)
      - 인터페이스 id
      - 인터페이스 명
      - 오퍼레이션 명
      - 오퍼레이션 개요
      - 사전 조건
      - 사후 조건
      - 파라미터
      - 반환 값
  - 정적,동적 모형 및 데이터 명세에 따른 인터페이스 설계서
    - 데이터 명세를 통한 인터페이스 설계서
  - 내부,외부 모듈 간 공통 기능 및 데이터 인터페이스 확인 방안
    - 외부, 내부 모듈 연계를 위한 인터페이스 기능 식별
    - 외부,내부 모듈 연계 방법 (**EAI, ESB 연계 방법**)
      - EAI(Enterprise Application Integration) 방식
        - 기업에서 운영되는 서로 다른 플랫폼 및 애플리케이션 간 정보를 전달, 연계, 통합이 가능하도록 해주는 솔루션
        - EAI 구축 유형 (**포허 메하**)
          - 포인트 투 포인트
          - 허브 앤 스포크
          - 메시지 버스
          - 하이브리드
      - ESB(Enterprise Service Bus) 방식
        - 기업에서 운영되는 서로 다른 플랫폼 및 애플리케이션들 간의 하나의 시스템으로 관리 운영할 수 있도록 서비스 중심의 통합을 지향하는 아키텍처
        - 느슨한 결합 방식으로 지원한다.

- 외부, 내부 모듈 연계를 위한 인터페이스 기능 식별 절차
- 외부 및 내부 모듈 간 인터페이스 데이터 표준 확인
  - 인터페이스 데이터 표준 확인
  - 송수신 시스템 간 인터페이스 데이터 표준 확인 절차

## chap2. 인터페이스 기능 구현

- 인터페이스 기능 구현 정의
  - 모듈 간 세부 설계서 확인
    - 컴포넌트 명세서
      - 컴포넌트의 개요, 내부 클래스의 동작, 인터페이스를 통해 외부와 통신하는 명세
    - 인터페이스 명세서
      - 컴포넌트 명세서에 명시된 인터페이스 클래스의 세부적인 조건 및 기능을 명시한 명세서
  - 일관되고 정형화된 인터페이스 기능 정의
    - 분석된 인터페이스의 기능, 데이터 표준, 모듈 설계 명세서를 통하여 일관되고 정형화된 인터페이스의 기능을 정의한다.
  - 정의된 인터페이스 기능에 대한 정형화
    - 정의된 인터페이스 기능을 표준화하고, 사람들이 보기 쉽게 정형화한다.
- 인터페이스 기능 구현
  - 사전에 정의된 기능에 대한 구체적 분석
  - 인터페이스 기능 구현 기술
    - **JSON**
      - 속성-값 쌍, 키-값 쌍 으로 이루어진 데이터 오브젝트를 전달하기 위해 인간이 읽을 수 있는 텍스트를 사용하는 개방형 표준 포맷
      - AJAX 에서 많이 사용되고 XML을 대체하는 주요 데이터 포맷.
      - 언어 독립형 데이터 포맷으로 다양한 데이터 프로그래밍 언어에서 사용된다.
      - 사람이 읽고 쓰기 용이, 기계가 분석하고 생성하기 용이함.
      - 숫자,문자열,배열,객체 자료형
      - 장점 : XML보다 가볍고 빠름, 자료 종류에 제한 X,
      - 단점 : 태그가 없어서 가독성 떨어짐.
    - **XML**
      - HTML의 단점을 보완한 인터넷 언어로 , SGML의 복잡한 단점을 개선한 특수한 목적을 갖는 마크업 언어.
      - 송,수신 시스템 간 데이터 연계의 편의성을 위해 전송되는 데이터 구조를 동일한 형태로 정의함.
      - 인간, 기계가 모두 이해할 수 있는 텍스트 형태로 마크업 포맷을 정의하기 위한 메타언어이다.
      - 사용자가 직접 문서의 태그를 정의할 수 있다. 사용자가 정의한 태그를 사용할 수 있다.
    - **AJAX** (Asynchronous Javascript And XML)
      - AJAX는 자바스크립트를 사용하여 웹 서버와 클라이언트 간 비동기적으로 XML 데이터를 교환하고 조작하기 위한 웹 기술이다.
      - HTML 만으로 어려운 다양한 작업을 웹 페이지에서 구현해서 이용자가 웹 페이지와 자유롭게 상호 작용할 수 있도록 구현하는 기법.
      - AJAX 주요 기술
        - XMLHttpRequest, JavaScript, XML, DOM, XSLT, HTML, CSS
    - **REST** (Representational State Transfer)
      - 웹과 같은 분산하이퍼미디어 환경에서 자원의 존재/상태 정보를 표준화된 HTTP 메서드로 주고 받는 웹 아키텍처이다.
      - REST 기본 형태 (리메메)

- 리소스(자원), 메서드(처리), 메시지 3가지로 구성
  - 행위에 대한 메서드
    - HTTP 메서드 중 CRUD에 해당하는 4가지 메서드만 사용한다.
      - POST (create)
      - GET (Select)
      - PUT (Update)
      - DELETE (Delete)
    - REST 리소스
  - REST 특징
    - 클라이언트/서버 구조
    - 무 상태성
    - 일관된 인터페이스
    - 캐시 처리 가능
    - 자체 표현 구조
- 인터페이스 구현
  - 데이터 통신을 사용하는 방법과 인터페이스 개체를 사용하여 인터페이스를 구현하는 방법이 있다.
  - 데이터 통신을 사용하는 인터페이스 구현
    - 인터페이스 객체 생성 구현
      - 인터페이스 객체를 생성하기 위해서 데이터베이스에 있는 정보를 SQL을 통해 선택한 후 이를 JSON으로 생성
    - 인터페이스 객체 전송 후 전송 결과를 수신 측에서 반환 받도록 구현
      - 송신 측에서 제이슨으로 작성된 인터페이스 객체를 AJAX 기술을 이용해 수신 측에 송신
  - 인터페이스 개체(Entity)를 사용하는 인터페이스 구현
    - 송신 시스템의 인터페이스 테이블
      - 송신 관련 정보를 관리하기 위한 항목과 송신 시스템에서 필요한 항목 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
      - 수신 측 시스템에서 인터페이스 데이터를 읽은 후 사전에 정의된 데이터 트랜잭션을 진행할 수 있도록 구현
  - 인터페이스 예외 처리 방안
    - 데이터 통신을 사용한 인터페이스에서 예외 처리 방법
      - 송신 측에서 예외 처리 방법
        - AJAX 호출 후 반환 값을 받아 어떻게 처리할지를 호출하는 부분에서 사전 정의
        - 반환 값은 성공과 실패로 나뉘며, 예외처리는 실패 시 발생
        - 404오류
      - 수신 측에서 예외 처리 방법
        - 수신 측에서 받은 JSON 객체를 처리 시에 try ~ catch 구문을 이용하여 발생한 예외를 처리하고 이를 송신 측에 전달
        - 별도 예외 결과를 설정하지 않아도 에러 발생 시 에러 결과 반환.
    - 인터페이스 개체를 사용하는 인터페이스에서 예외 처리 방법
      - 송신 인터페이스 테이블에서 예외 처리 방법
      - 수신 인터페이스 테이블에서 예외 처리 방법
        - 책에 있는 표!! 보면서 외워보자
  - 인터페이스 보안 기능 적용
    - 인터페이스 보안 취약점

- 데이터 통신 시 데이터 탈취 위험
  - 스니핑을 통해 데이터 전송 내역을 감청하여 데이터를 탈취하는 위협이 존재한다.
- 데이터 통신 시 데이터 위,변조 위험
  - 전송 데이터에 대한 삽입, 삭제, 변조 공격을 통한 시스템 위협 존재.
- 인터페이스 보안 구현 방안
  - 시큐어 코딩 가이드 적용
    - (입보시 예코캡아)
    - 입력데이터 검증 및 표현
    - 보안 기능
    - 시간 및 상태
    - 에러 처리
    - 코드 오류
    - 캡슐화
    - API 오용
  - 데이터베이스 보안 적용
    - 데이터베이스 암호화 알고리즘 (대비해)
      - 대칭 키 암호화 알고리즘
        - 암호,복호화에 같은 암호 키를 쓰는 알고리즘 의미
      - 비대칭 키 암호화 알고리즘
        - 공개키는 누구나 알 수 있지만, 대응하는 비밀키는 키의 소유자만이 알 수 있도록 함.
      - 해시 암호화 알고리즘
        - 해시값으로 원래 입력값을 찾아낼 수 없는 일방향성의 특성을 가진 알고리즘.
    - 데이터베이스 암호화 기법
      - (애플티하)
      - API 방식
      - Plug-in
      - TDE 방식
      - Hybrid 방식

---

## chap3. 인터페이스 구현 검증

- 인터페이스 구현 검증 도구
  - 구현된 인터페이스의 동작을 검증하기 위해 인터페이스 구현 및 감시 도구를 통해서 인터페이스 동작 상태를 검증하고 모니터링 할 수 있다.
  - 인터페이스 구현 검증 도구 종류 (엑스피 엔셀웨)
  - xUnit
  - STAF
  - FitNesse
  - NTAF
  - Selenium
  - watir
- 인터페이스 오류 처리 확인 및 보고서 작성
  - 인터페이스 오류 처리 방법
  - (화로테)

- 사용자 **화면**에서 오류를 인지하도록 구현
- 인터페이스 오류 **로그** 생성
- 인터페이스 관련 **테이블**에 오류 사항 기록