

인터페이스 구현

(응용SW개발)

외부 및 내부 모듈 간 인터페이스
데이터 표준 확인

학습 내용

- 인터페이스 데이터 표준
- 외부 및 내부 모듈 간 인터페이스 데이터 표준 확인

학습 목표

- 인터페이스 데이터 표준 정의를 설명하고, 표준 형태를 사용할 수 있다.
- 외부 및 내부 모듈 간 인터페이스 데이터 표준을 설명할 수 있다.

1. 인터페이스 데이터 표준

1 인터페이스 데이터 표준 개념

◆ 인터페이스 데이터 표준 개념

인터페이스
데이터 표준

인터페이스를 위해 인터페이스가 되어야 할
**범위의 데이터 형식이나 규칙, 또는
전송 방법들을 정의**하는 것

- ◆ 기존 데이터 중 공통 영역을 추출하여 정의
- ◆ 인터페이스를 위해 한쪽의 데이터를 변환

외부 및 내부 모듈 간 데이터 교환 및 상호 호환을 위해
인터페이스 데이터 표준을 정의, 관리해야 함

1. 인터페이스 데이터 표준

1 인터페이스 데이터 표준 개념

◆ 인터페이스 데이터 표준 개념

인터페이스 표준 선정 시 고려 요소

1 최소의 노력으로 데이터 변환이 가능한 것

데이터 변환 작업이 적은, 즉 많은 시스템이 사용하고 있는 인터페이스를 표준으로 정의

2 통신 처리 오버헤드, 속도 등을 고려한 최적의 기법

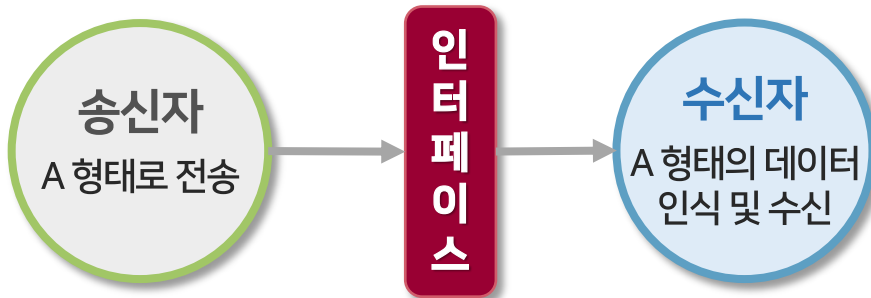
예 | 문자형식으로 보내는 통신 패킷을 바이너리 형식으로 변환하여 패킷 크기를 줄이고 인터페이스 작업을 함

1. 인터페이스 데이터 표준

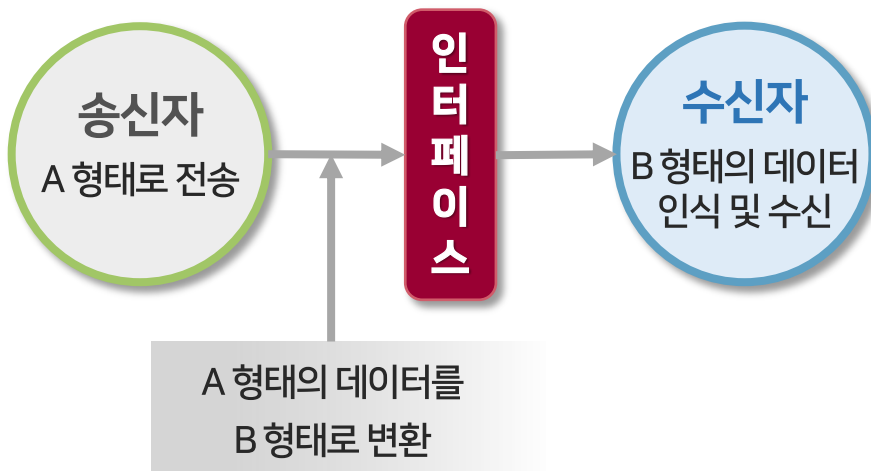
1 인터페이스 데이터 표준 개념

❖ 인터페이스 데이터 표준 개념

인터페이스 표준이 일치하여
전달될 데이터 변환이 필요 없는 경우



인터페이스 표준이 맞지 않아
데이터 변환이 필요한 경우
(B 형태를 표준으로 정의한 경우)



1. 인터페이스 데이터 표준

1 인터페이스 데이터 표준 개념

◆ 인터페이스 데이터 표준 예시

- ◆ 데이터들의 형식이나 규칙을 찾는 예시
- ◆ 상호 인터페이스 해야 할 모듈의 데이터 표준을 식별, 작성할 경우 인터페이스 데이터 표준도 같이 정의하고 산출물에 표현
- ◆ 일반적으로 업무 테이블 정의서 내용과 같지만, 인터페이스가 필요한 부분은 별도 표기하여 인터페이스에 사용될 것임을 알려 줌

예 | 데이터베이스 테이블 정의서에서 인터페이스 송신 측의 데이터 표준 작성

프로젝트명	그룹	ASP 전자세금계산서 CLINET I/F	문서명	테이블정의서			작성자		작성일자	
시스템명		전자세금계산서					승인자		PAGE	
TABLE한글명	세금계산서정보 그룹 ASP 전송 마스		TABLE영문명							
항목명	영문명		KEY	TYPE	길이	NULL	비고			
F_INVO_NUM	세금계산서번호(I/F)		PK	VARCHAR	24	N	기간제 유니크한 번호 부여 (각 그룹사별 코드화 하여 사용, 중복제하는 그룹 ASP에서 별도로 관리함) ex) 20090930-0000-0000000001			
F_INVO_NUM_REF	전자세금계산서 부모번호(I/F)(수정경우)			VARCHAR	24		수정(세금)계산서의 원 세금계산서번호(I/F) → 수정(세금)계산서에 대한 원 세금계산서 번호로 입력 시 그룹 ASP에서 수정세금계산서에 대한 원 세금계산서 링크링 → 입력 안해도 무방함.			
TAX_APPROVAL_NO	전자세금계산서 승인번호			VARCHAR	24	N	작성년월일(8자리)+국세청승인번호(ASP 승인번호-8자리)+그룹사승인코드(2자리)+일련번호6자리 작성년월일 : 세금계산서 작성년월일 (하기의 TAX_ISSUE_DT 와 동일) 국세청 승인번호 : 8자리로 ASP 에서 국세청 승인 후 백포 예정 그룹사 승인코드 : 그룹사코드 Sheet 참조 주의사항 : 시스템별로 중복되면 안됨			

1. 인터페이스 데이터 표준

1 인터페이스 데이터 표준 개념

◆ 인터페이스 데이터 표준 예시

- ◆ 인터페이스에 필요한 데이터 포맷의 표준을 표현
- ◆ Json, DB, XML 등 인터페이스 모듈의 표준 형태는 다양한 형태로 표현 가능

예 | 인터페이스 모듈의 데이터 표준 표현

프로젝트명	ASP 전자세금계산서 CLINET I/F	문	테이블정의서				작성자		작성일자	
시스템 명	전자세금계산서	서					승인자		PAGE	
TABLE한글명	상태변경 처리요청 세금계산서 (이력포)	TABLE 영문명								
항목명	영문명	KEY	TYPE	길이	NULL					비고
IF_INVO_NUM	기간계 세금계산서번호(I/F)	PK	VARCHAR	24	N					
TAX_STATE_REQ_CREATE_DT	상태변경처리요청 등록일시	PK	DATETIME		N					
TAX_ISSUE_DT	작성일자(세금계산서 내 작성년월일)		VARCHAR	8	N					YYYYMMDD형식
TAX_STATE_REQ_CD	상태변경요청 코드		CHAR	3	N					그룹 ASP 상태값 변경 요청 코드 입력 세금계산서 폐기 : 400 (ERP폐기요청) 세금계산서 재전송 : 250 (반려기각) 매일 재전송 : 900 SMS 재전송 : 910
SEND_PROC_DT	전송처리 일시		DATETIME							전송 처리 후 입력되는 필드

1. 인터페이스 데이터 표준

2 인터페이스 표준 형태

❖ 길이 고정(지정)방식 인터페이스 표준

- ❖ 데이터의 전송 형식을 규정하는 표준의 예시
- ❖ A와 B사이의 데이터 전송 수신하는 인터페이스 표준에 있어서 일정한 크기를 기준으로 데이터의 내용을 구분 짓는 방식

예 | 모바일 시스템에서 상품 코드와 생산 연도를 보내면 재고관리 시스템에서 상품명, 소비자가격, 재고 수량, 재고 위치 주소를 제공하는 인터페이스 서비스를 생각해 봄



1. 인터페이스 데이터 표준

2 인터페이스 표준 형태

❖ 길이 고정(지정)방식 인터페이스 표준

입력 값 '9876520200523'이라고 보냈을 때
'고릴라 발톱0002500000010경기도제2창고12-23지역'
이라고 응답을 받았다고 가정



이미 정해 놓은 인터페이스 표준에 따라
상품 코드가 98765, 생산 연도 2020년 5월 23일인
상품의 재고를 찾겠다는 모바일 시스템의 요청



재고 관리 시스템은 해당 상품이 고릴라 발톱이며
소비자가격은 25000원, 10개의 재고가
경기도 제2창고의 12-23 지역에 있음을 알려 준 것

1. 인터페이스 데이터 표준

2 인터페이스 표준 형태

◆ 길이 고정(지정)방식 인터페이스 표준

장점

길이 고정 방식의 인터페이스 표준은 쉽게 이해되고, 구현 가능하기 때문에 가장 많이 활용

단점

서로의 인터페이스 내용이 변경되는 경우 수정이 번거로움

- ◆ 송수신 측에서 서로의 인터페이스 표준(구성 정보)을 관리하고 있어야 함



1. 인터페이스 데이터 표준

2 인터페이스 표준 형태

◈ 가변 길이 방식 인터페이스 표준 예시 (CSV)

가변 길이 방식
인터페이스
표준

A와 B사이의 데이터 전송 수신하는 인터페이스
표준에서 일정한 크기를 기준으로
데이터의 내용을 구분 짓는 방식의 단점을
보완하기 위해 **필드의 구분자를 두는 등의
방식으로 데이터를 전송하는 것**

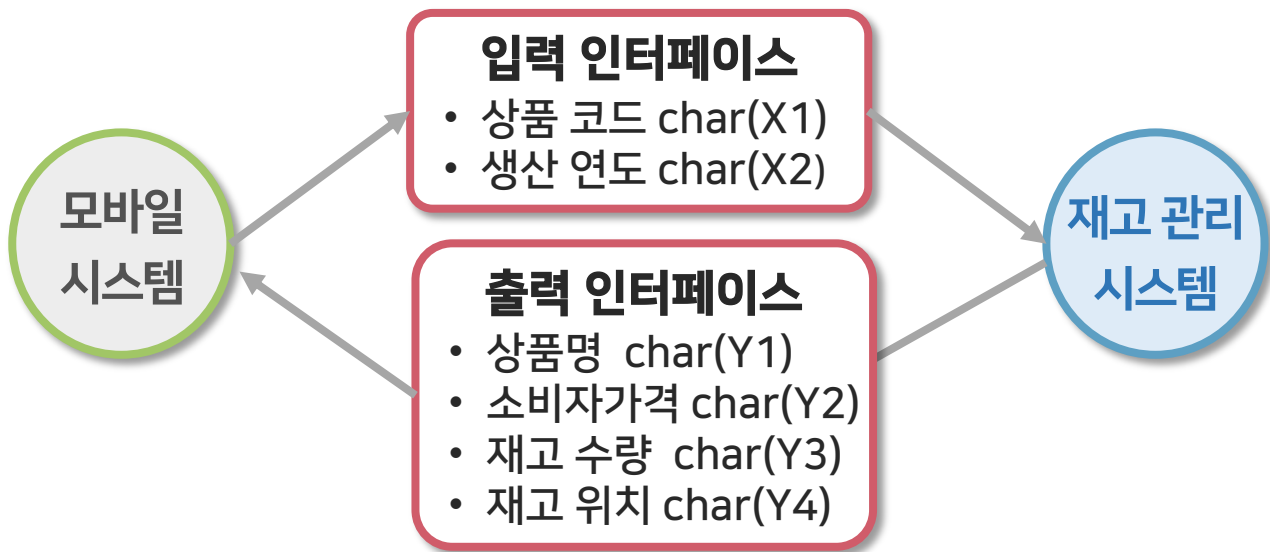
바나나,00025000,00010,경기도제2창고12-23지역	
사과,00025000,00010,서울제1창고1-3지역	

1. 인터페이스 데이터 표준

2 인터페이스 표준 형태

◆ 가변 길이 방식 인터페이스 표준 예시 (CSV)

모바일 시스템에서 상품 코드와 생산 연도를 보내면 재고관리
예 | 시스템에서 상품명, 소비자가격, 재고 수량, 재고 위치 주소를
제공하는 인터페이스 서비스를 생각해 봄



◆ 각 항목별 구분자를 두어서 전송하는 방법을 사용할 수 있음



- CSV(Comma Separated Value)
- 많이 사용되는 방법
- [,](콤마)를 구분자로 전송하는 것

1. 인터페이스 데이터 표준

2 인터페이스 표준 형태

◈ 가변 길이 방식 인터페이스 표준 예시 (CSV)

입력 값을 '98765,20200523'이라고 보냈을 때
'고릴라발톱,00025000,00010,경기도제2창고12-23지역' 이라고 응답을 보낸다면 각 항목은 콤마(,)를
기준으로 구분하여 추출

CSV 인터페이스 표준에 따라 상품 코드가 98765,
생산 연도 2020년 5월 23일인 상품의 재고를 찾겠다는
모바일 시스템의 요청

재고 관리 시스템은 해당 상품이 고릴라 발톱이고
소비자 가격은 25000원이며 10개의 재고가
경기도 제2 창고의 12-23 지역에 있음을 알려줌

1. 인터페이스 데이터 표준

2 인터페이스 표준 형태

◈ 가변 길이 방식 인터페이스 표준 예시 (CSV)

항목 안에 콤마가 있다면 문제가 생길 수 있음

예 | 해당 상품가격 25000원을 돈의 단위이기 때문에
콤마를 붙여서 25,000으로 전송하였다면,
해당 인터페이스 내용이 달라질 수 있음



이런 경우 복잡한 구분자를 사용

예 | 구분자를 "^_@_^" 이와 같이 복잡하게 사용하면
필드 내에서 내용이 중복될 확률은 적음

고정 길이 인터페이스 표준과 마찬가지로
인터페이스 필드 정보는 상호시스템에서
모두 가지고 있어야 함

1. 인터페이스 데이터 표준

2 인터페이스 표준 형태

◈ 가변 길이 방식 인터페이스 표준 예시 (XML)

XML

- eXtensible Markup Language의 약어
- W3C에서 여러 특수 목적의 **마크업 언어**를 만드는 용도에서 권장되는 다목적 **마크업 언어**

- 태그 등을 이용하여 데이터 구조를 기술하는 언어의 한 가지
- 가장 친숙하고 흔하게 접할 수 있는 마크업 언어로 HTML이 있음



1. 인터페이스 데이터 표준

2 인터페이스 표준 형태

◆ 가변 길이 방식 인터페이스 표준 예시 (XML)

XML 특징

- 1 어디부터 어디 까지가 데이터 항목 이름이고, 실제 데이터이며, 데이터 단위 인지 표현 가능
- 2 데이터에 의미를 부여하는 메타데이터 기술 가능

예 | `<cpumodel>Intel i7-8750H</cpumodel>`과 같이
메타데이터의 태그체 사이에 데이터를 넣어주는
형식으로 표현

- 3 원래는 웹 사용 목적으로 만들었는데, 웹 환경이 아닌 일반 TCP/IP 네트워크 통신을 할 때도 점점 사용빈도 증가
- 4 고정 길이 인터페이스의 단점을 커버해줄 수 있는 표준
- 5 보내고 받는 데이터의 인터페이스 규칙인 메타 데이터가 포함되어 있기 때문에 송·수신자 간의 인터페이스 규칙을 미리 알지 않더라도 인터페이스 처리 가능

1. 인터페이스 데이터 표준

2 인터페이스 표준 형태

❖ 가변 길이 방식 인터페이스 표준 예시 (XML)

- ❖ A와 B사이에서 데이터를 전송·수신하는 인터페이스 표준에 XML을 사용할 경우 데이터의 형식(메타데이터)과 데이터 내용을 같이 전송하는 개념

예 | 모바일 시스템에서 상품 코드와 생산 연도를 보내면 재고관리 시스템에서 상품명, 소비자가격, 재고 수량, 재고 위치 주소를 제공하는 인터페이스 서비스



1. 인터페이스 데이터 표준

2 인터페이스 표준 형태

◈ 가변 길이 방식 인터페이스 표준 예시 (XML)

XML형식의 인터페이스
표준으로 상품 코드가 98765,
생산 연도는 2020년 5월
23일인 상품의 재고를
찾겠다는 것을 모바일 시스템에
요청

```
<XML>
<상품코드>98765</상품코드>
<생산년도>20200523</생산년도>
>
</XML>
```



재고 관리 시스템은 해당
상품이 고릴라 발톱,
소비자가격은 25,000원,
10개의 재고가 경기도 제2
창고의 12-23 지역에
있음을 알려 주는 경우

```
<XML>
<상품명>고릴라발톱</상품명>
<소비자가격>25,000</소비자가격>
<재고수량>10</재고수량>
<재고위치>경기도 제2 창고의 12-
23지역</재고위치>
</XML>
```

1. 인터페이스 데이터 표준

2 인터페이스 표준 형태

◈ 가변 길이 방식 인터페이스 표준 예시 (JSON)

JSON

- 서버에서 클라이언트로 데이터를 보낼 때 가장 많이 사용하는 양식
- XML이 헤더와 태그 등의 여러 요소로 가독성이 떨어지고, 쓸데없이 용량을 잡아먹는 단점을 해결하기 위해 간결하고 통일된 양식으로 인터페이스를 표현하는 방식

클라이언트가 사용하는 언어에 관계 없이
통일된 데이터를 주고받을 수 있도록,
일정한 패턴을 지닌 문자열을 생성해 내보냄



클라이언트가 해석하여 데이터를 자기만의 방식으로
온전히 저장, 표시할 수 있게 됨

1. 인터페이스 데이터 표준

2 인터페이스 표준 형태

❖ 가변 길이 방식 인터페이스 표준 예시 (JSON)

- ❖ 웹 표준(W3C의 표준) = XML
- ❖ JSON은 2009년 말 Ecma에 의해 ECMAScript5에서 스크립트 엔진의 기본 기능으로 내장됨
- ❖ 새롭게 출시되는 브라우저들 기준에서는 DOM을 통해 XML 파싱하는 것보다 JavaScript 엔진에서 JSON을 메모리로 받는 쪽이 성능으로 보나 트래픽으로 보나 훨씬 더 나은 방법이 되었음



현재는 XML보다 더 많은 표준으로 채택, 사용 중



A와 B사이에 데이터를 전송·수신하는 인터페이스 표준

- ❖ JSON을 사용한다면 데이터의 형식(메타데이터)과 데이터 내용을 같이 전송하는 개념은 동일함
- ❖ 다중 입력을 전송, 다중의 데이터를 받는 구조라면 **JSON이 훨씬 유리한 표준이 됨**

1. 인터페이스 데이터 표준

2 인터페이스 표준 형태

◆ 가변 길이 방식 인터페이스 표준 예시 (JSON)

모바일 시스템에서 상품 코드와 생산 연도를 보내면 재고관리
예 | 시스템에서 상품명, 소비자 가격, 재고 수량, 재고 위치 주소를
제공하는 인터페이스 서비스

입력 인터페이스

메타데이터	데이터 내용 (1)	데이터 내용 (2)	데이터 내용 (n)
• 상품 코드	• char(X1)	• char(X1) ...	• char(X1)
• 생산 연도	• char(X2)	• char(X2)	• char(X2)

모바일
시스템

재고 관리
시스템

출력 인터페이스

메타데이터	데이터 내용 (1)	데이터 내용 (2)	데이터 내용 (n)
• 상품명	• char(Y1)	• char(Y1)	• char(Y1)
• 소비자 가격	• char(Y2)	• char(Y2) ...	• char(Y2)
• 재고 수량	• char(Y3)	• char(Y3)	• char(Y3)
• 재고 위치	• char(Y4)	• char(Y4)	• char(Y4)

1. 인터페이스 데이터 표준

2 인터페이스 표준 형태

◆ 가변 길이 방식 인터페이스 표준 예시 (JSON)

XML을 인터페이스
표준으로 사용하여
여러 개의 입력으로
정보를 얻어오는 경우

```
<XML>
<input1>
<상품코드>98765</상품코드>
<생산년도>20200523</생산년도>
</input1>
~~~~~
<input3>
<상품코드>98767</상품코드>
<생산년도>20200723</생산년도>
</input3>
</XML>
```



반복적인
메타데이터에 의해
많은 용량이 소요됨

```
<XML>
<output1>
<상품명>고릴라발톱</상품명>
<소비자가격>25,000</소비자가격>
</output1>
~~~~~
<output3>
<상품명>고양이발톱</상품명>
<소비자가격>35,000</소비자가격>
</output3>
</XML>
```

1. 인터페이스 데이터 표준

2 인터페이스 표준 형태

◈ 가변 길이 방식 인터페이스 표준 예시 (JSON)

JSON 장점

- XML사용의 단점 개선
- 보다 직관적인 인터페이스 작성 가능
- Java계열 실 프로그램에서도 해석기에서의 성능이 우수함

```
{
  "fieldname":["상품코드","생산일자"],
  "data":[
    {"98765","20200523"},
    {"98766","20200623"},
    {"98767","20200723"}
  ]
}
```

```
{
  "fieldname":["상품명","소비자가격","재고수량","재고위치"],
  "data":[
    {"고릴라발톱","25,000","10","경기도 제2 창고의 12-23 지역"},
    {"강아지발톱","15,000","120","경기도 제1 창고의 11-22 지역"},
    {"고양이발톱","35,000","70","경기도 제3 창고의 33-44 지역"}
  ]
}
```

2. 외부 및 내부 모듈 간 인터페이스 데이터 표준 확인

1 데이터 표준 확인 절차

🔹 데이터 표준 확인 절차

1 입·출력값의 의미 분석

- 데이터의 특성 등 참고할 만한 사항을 구체적으로 작성

2 데이터 표시 기준 및 방법 확정

- 해당 항목에 대하여 데이터를 어떤 형식으로 보낼 것인지 구체적으로 정의
- 예를 들어 일자를 YYYYMMDD로 작성할 것인지, 항목에 사용될 코드 값은 어떻게 정의할 것인지 확정

2. 외부 및 내부 모듈 간 인터페이스 데이터 표준 확인

1 데이터 표준 확인 절차

❖ 데이터 표준 확인 절차

3 식별된 데이터 항목 검증

- 도출된 항목이 인터페이스를 위한 필수 항목인지, 더 필요한 항목이 있는지 검증
- 식별된 인터페이스 기능을 통해 인터페이스 데이터 항목 식별
- 인터페이스 기능을 구현하기 위해 필요한 항목이 무엇인지 분석, 나열하는 형태로 작성
- 필요 데이터 항목과 이전에 식별된 데이터 인터페이스 항목에서 조정(수정, 추가, 삭제)되어야 할 항목은 없는지 검토

4 데이터 인터페이스 및 인터페이스 기능을 분리하고 최종 도출된 데이터 표준(데이터 형식이나 규칙) 확인

- 데이터 인터페이스 및 인터페이스 기능을 통해 필요한 데이터 표준 및 조정해야 할 항목을 검토, 확인
- 이에 따라 인터페이스 데이터 표준을 최종적으로 확인
- 확인된 데이터 표준을 어디에서 도출(데이터 인터페이스, 인터페이스 기능)하였는지 구분하여 작성

2. 외부 및 내부 모듈 간 인터페이스 데이터 표준 확인

2 데이터 표준 확인 사례

입·출력 값의 의미 분석

- 식별된 데이터 인터페이스의 입력 값, 출력 값이 의미하는 내용을 파악
- 각 출력 값이 의미하는 바와 데이터 특성 등 참고할 만한 사항을 구체적으로 작성

구분	주요기능	시나리오
입력 값	급여 코드	<ul style="list-style-type: none"> 급여 작업의 단위를 나타내는 Key 값 <p>예 2018년 6월 정규직 급여: 201806R001</p>
	급여 일자	<ul style="list-style-type: none"> 급여를 지급하는 일자(인터페이스 시 회계 일자로 활용)
	급여 계산 결과	<ul style="list-style-type: none"> 각 직원 별 급여 계산 결과 총 금액(급여, 상여 구분), 공제 금액(소득세, 주민세, 국민연금, 건강 보험, 고용 보험, 사내 공제 금액 구분), 직원 계좌 정보
	전표 계정 정보	<ul style="list-style-type: none"> 각 항목별 전표에 들어가야 할 계정 (급여 계정, 건강 보험료 계정, 소득세/주민세 계정 등)

2. 외부 및 내부 모듈 간 인터페이스 데이터 표준 확인

2 데이터 표준 확인 사례

데이터 표시 기준이나 방법 확정

- 해당 항목에 대하여 데이터를 어떤 형식으로 보낼 것인지 구체적으로 정의

예 | 일자를 YYYYMMDD로 작성할 것인지,
항목에 사용될 코드 값은 어떻게 정의할 것인지 확정

구분	필드 구분 (변수)	데이터 인터페이스 기준
입력 값	급여 코드	<ul style="list-style-type: none"> 정규직은 R, 계약직은 T 예 201806R 3자리 숫자로 급여 코드 증가 시 숫자를 증가시킴 해당 월에 처음 부여된 코드는 1번을 부여 예 201806R001
	급여 일자	<ul style="list-style-type: none"> YYYYMMDD 형태의 8자리로 표현 예 20180605
	급여 계산 결과	<ul style="list-style-type: none"> 각 직원별 급여 계산 결과 항목 <ul style="list-style-type: none"> - 사번, 근무 일수, 소속, 직급, 급여, 상여, 비과세 급여, 총 지급 금액, 소득세, 주민세, 국민연금, 건강 보험, 고용 보험, 회사 공제 금액, 총 공제 금액, 실지급여 데이터 포맷은 엑셀로 차후 정의

2. 외부 및 내부 모듈 간 인터페이스 데이터 표준 확인

2 데이터 표준 확인 사례

데이터 표시 기준이나 방법 확정

구분	필드 구분 (변수)	데이터 인터페이스 기준
입력 값	전표 계정 정보	<ul style="list-style-type: none"> 급여 계정, 상여 계정, 소득세, 주민세, 국민연금, 건강보험, 고용 보험, 회사 공제 금액 계정, 지급 은행 코드, 지급 계좌 각 계정은 회사 회계 계정에 맞게 코드화 입력 예 [급여 계정: ABC3343F(예시 코드)] 은행 코드는 임의 자리로 정의 예 XX 은행: KBY 지급 계좌는 - 없이 숫자로 입력

2. 외부 및 내부 모듈 간 인터페이스 데이터 표준 확인

2 데이터 표준 확인 사례

◆ 식별된 데이터 항목 검증

식별된
데이터 항목
검증

도출된 항목이 인터페이스를 위한 필수 항목인지,
더 필요한 항목이 있는지 검증하는 것

- ◆ 식별된 인터페이스 기능으로 인터페이스 데이터 항목 식별
- ◆ 인터페이스 기능을 구현하려면 필요한 항목이 무엇인지 분석하고 나열하는 형태로 작성
- ◆ 필요 데이터 항목과 이전에 식별된 데이터 인터페이스 항목에서 조정(수정, 추가, 삭제)되어야 할 항목은 없는지 검토

2. 외부 및 내부 모듈 간 인터페이스 데이터 표준 확인

2 데이터 표준 확인 사례

❖ 식별된 데이터 항목 검증

인터페이스 기능	인터페이스 기능 수행을 위한 필요 데이터 항목	기 식별된 데이터 인터페이스 항목
전표 발생을 위한 필수 입력 값을 외부 모듈로부터 수신함	필수 입력 값 • 전표 일자, 계정, 금액 • 발의 부서, 귀속 부서 • 거래처, 지급 여부	<ul style="list-style-type: none"> • 급여 코드, 급여 일자, 급여 계산 결과 전표 • 계정 정보 (발의, 귀속 부서는 급여 결과에 포함되고 모두 지급되는 항목이며, 직원 거래처는 이미 등록 완료) • 추가될 항목 없음
회계 시스템으로 전표 발생을 요청함		
회계 시스템에서 발생된 전표 발생 작업 결과를 수신 받음	필수 수신 값 • 전표 발생 여부 • 차, 대변 검증 여부 • 부서 별 금액, 계정 별 금액 • 총 지급, 공제 금액	전표 일반 정보 • 전표 금액 • 차, 대변 금액 • 거래처 정보 (대부분 정보가 들어가 있으나 전표 차·대변 검증 결과는 추가 필요)

2. 외부 및 내부 모듈 간 인터페이스 데이터 표준 확인

2 데이터 표준 확인 사례

- ❖ 데이터 인터페이스 및 인터페이스 기능 분리,
최종 도출된 데이터 표준(데이터들의 형식이나 규칙) 확인

데이터 인터페이스 및 인터페이스 기능을 통해
필요한 데이터 표준 및 조정해야 할 항목 검토 및 확인



인터페이스 데이터 표준을 최종 확인



확인된 데이터 표준을 어디에서 도출(데이터
인터페이스, 인터페이스 기능)하였는지 구분 및 작성

2. 외부 및 내부 모듈 간 인터페이스 데이터 표준 확인

2 데이터 표준 확인 사례

- ❖ 데이터 인터페이스 및 인터페이스 기능 분리,
최종 도출된 데이터 표준(데이터들의 형식이나 규칙) 확인

구분	필드 구분 (변수)	데이터 인터페이스 기준	확인 출처
입력 값	급여 코드	<ul style="list-style-type: none"> 급여 지급 연월을 숫자 6자리로 명시 예 201806 정규직은 R, 계약직은 T 예 201806R 3자리 숫자로 급여 코드 증가 시 숫자를 증가 해당 월에 처음 부여된 코드는 1번을 부여 예 201806R001 	데이터 인터페이스
	급여 일자	<ul style="list-style-type: none"> YYYYMMDD 형태의 8자리로 표현 예 20180605 	데이터 인터페이스
출력 정보	차·대 변 검증	<ul style="list-style-type: none"> 계정 별, 부서 별 항목 수와 총 금액 검증 예 부서별 검증 급여 계산 값: 부서 XXX 인원 ○○명 금액 3333 전표 발생: 부서 XXX 인원 ○○명 금액 3333 XXX 부서 검증 결과: 정상 발생 	인터페이스 기능
	거래처 정보	<ul style="list-style-type: none"> 거래처(직원): 사번_이름 형태로 정의 계좌 정보: 은행 코드_계좌 번호로 정의 	데이터 인터페이스

핵심요약

인터페이스 데이터 표준

- 외부 및 내부 모듈 간 데이터를 교환하고 상호 호환을 위해 **인터페이스 데이터 표준을 정의하고 이를 관리**해야 함

- **인터페이스 데이터 표준** 인터페이스를 위해 인터페이스가 되어야 할 **범위의 데이터 형식이나 규칙, 또는 전송 방법들을 정의하는 것**

■ 인터페이스 표준의 사례

- ① 고정된 길이로 필드를 구분하는 방법
- ② CSV와 같이 필드 구분자로 필드를 구분하는 방법
- ③ 필드의 정보인 메타데이터와 데이터를 같이 태그 형식으로 보내는 XML, JSON

핵심요약

외부 및 내부 모듈 간 인터페이스 데이터 표준 확인

■ 외부 및 내부 모듈 간 인터페이스 데이터 표준을 확인하는 절차

- ①입·출력값의 의미 분석
- ②데이터 표시 기준이나 방법 확정
- ③식별된 데이터항목 검증
- ④데이터 인터페이스 및 인터페이스 기능 분리
- ⑤최종 도출된 데이터 표준 확인