

인터페이스 구현

(응용SW개발)

인터페이스 구현



한국기술교육대학교
온라인평생교육원

학습 내용

- 인터페이스 구현 도구
- 인터페이스 구현 절차

학습 목표

- 인터페이스 구현 도구에 대해 이해하고 설명할 수 있다.
- 인터페이스 구현 절차에 대해 이해하고 설명할 수 있다.

1. 인터페이스 구현 도구

1 데이터 통신을 통한 인터페이스 구현

❖ 데이터 통신을 통한 인터페이스 구현

인터페이스 구현 방법

데이터 통신을 이용한 방법

인터페이스 테이블을 이용한 방법

데이터 통신을 이용한 인터페이스 구현 방법이란?

애플리케이션 영역에서 인터페이스 형식에 맞춘
데이터 포맷을 인터페이스 대상으로 전송하고
이를 수신 측에서 파싱(Parsing)하여 해석하는 방식

1. 인터페이스 구현 도구

1 데이터 통신을 통한 인터페이스 구현

❖ 데이터 통신을 통한 인터페이스 구현

- ❖ 주로 JSON 및 XML 형식의 데이터 포맷을 사용하여 인터페이스 구현
- ❖ 그 외 통신처리속도 및 서비스 성격에 따른 다양한 방법
 - TCP/IP의 Socket-API를 사용하는 방식
 - Broadcasting 방식
 - X.25 등 전통적 통신방식
- ❖ 인터넷 통신 방식이 아닌 RFID, 블루투스 등의 방식을 사용하는 경우도 있음

1. 인터페이스 구현 도구

1 데이터 통신을 통한 인터페이스 구현

JSON(JavaScript Object Notation)



JSON

속성-값 쌍(Attribute-Value Pairs)으로 이루어진 데이터 오브젝트를 전달하기 위해 사용하는 개방형 표준 포맷

- ◆ AJAX(Asynchronous JavaScript And XML)에서 많이 사용되고 XML을 대체하는 주요 데이터 포맷
- ◆ 언어 독립형 데이터 포맷
- ◆ 다양한 프로그래밍 언어에서 사용

1. 인터페이스 구현 도구

1 데이터 통신을 통한 인터페이스 구현

JSON(JavaScript Object Notation)

JSON의 기본 자료형

구분	예시	설명
수 (Number)	<ul style="list-style-type: none"> 정수: 75, 1974, -114 실수(고정 소수점): 3.14, -2.712 실수(부동 소수점): 1e4, 2.5e12 	<ul style="list-style-type: none"> 기본 자료형 8진수나 16진수 표현 방법은 지원하지 않음
문자열 (String)	"1234", "Lovd", "문자", "₩", "json₩"	<ul style="list-style-type: none"> 큰 따옴표로 묶어야 함 ₩는 특수 기호 문자를 표현하기 위해 사용(₩, 탭 등)
배열 (Array)	[10, {"v": 20}, [30, "마흔"]]	<ul style="list-style-type: none"> 배열은 대괄호([])로 나타냄 배열의 각 요소는 기본 자료형, 배열, 객체 각 요소는 쉼표(,)로 구분
객체 (Object)	{"name2": 50, "name3": "값3", "name1": true}	<ul style="list-style-type: none"> 객체는 이름 : 값 쌍의 집합으로 중괄호({ })를 사용 이름은 문자열이기 때문에 반드시 따옴표로 표현 값은 기본 자료형임

1. 인터페이스 구현 도구

1 데이터 통신을 통한 인터페이스 구현

📦 JSON(JavaScript Object Notation) 사용 예시

예 | JSON을 사용하여 데이터 전송 포맷을 만든 예제

```
{
    "이름": "홍길동",
    "나이": 38,
    "성별": "남",
    "주소": "경기도 용인시 기흥구 중동",
    "특기": ["프로그래밍", "영어"],
    "가족관계": {"#": 3, "아내": "전지현", "딸": "김고은"},
    "회사": "서울시 강남구 논현동"
}
```

1. 인터페이스 구현 도구

1 데이터 통신을 통한 인터페이스 구현

JSON(JavaScript Object Notation) 사용 예시

예 | JSON을 작성하는 프로그램 예제

```
import java.io.*;
import org.json.simple.*;

public class JavaJSON {
    public static void main(String[] args) {
        //Json 객체 생성.
        JSONObject jobj = new JSONObject();
        jobj.put("name", "iamhpd");
        jobj.put("company", "ktech");
        jobj.put("age", "33");

        //Json Array 생성.
        JSONArray maglist = new JSONArray();
        maglist.add("JSON test");
        maglist.add("hello JSON");

        jobj.put("list", maglist);
    }
}
```

원하는 출력결과를 JSON문법에 맞도록 파일에 기록



Create JSON Object :
{"age":"33","company":"ktech","name":"iamhpd","list":["JSON test","hello JSON"]}

1. 인터페이스 구현 도구

1 데이터 통신을 통한 인터페이스 구현

🔗 JSON(JavaScript Object Notation) 사용 예시

예 | JSON을 작성하는 프로그램 예제

```
        try {  
            FileWriter file = new  
FileWriter("c:\\myJson.json");  
            file.write(jobj.toJSONString());  
            file.flush();  
            file.close();  
        } catch (IOException e) {  
            e.printStackTrace();  
        }  
        System.out.println("Create JSON Object : "+jobj);  
    }  
}
```

원하는 출력결과를 JSON문법에 맞도록 파일에 기록

1. 인터페이스 구현 도구

1 데이터 통신을 통한 인터페이스 구현

📦 JSON(JavaScript Object Notation) 사용 예시

예 | JSON을 수신하는 프로그램 예제, JSON 내용을 읽고 해석

```
import java.io.*;
import org.json.simple.*;
import org.json.simple.parser.*;
import java.util.Iterator;

public class JavaJSON_parse {
    public static void main(String[] args) {
        JSONParser parser = new JSONParser();

        try {
            // myJson.json파일을 읽어와 Object로 파싱
            Object obj = parser.parse(new
FileReader("c:\\myJson.json"));

            JSONObject jsonObject = (JSONObject) obj;

            String name = (String) jsonObject.get("name");
            String age = (String) jsonObject.get("age");
            String company = (String)
jsonObject.get("company");
```

1. 인터페이스 구현 도구

1 데이터 통신을 통한 인터페이스 구현

🔹 JSON(JavaScript Object Notation) 사용 예시

예 | JSON을 수신하는 프로그램 예제, JSON 내용을 읽고 해석

```

        System.out.println("name:" + name);
        System.out.println("age:" + age);
        System.out.println("company:" + company);

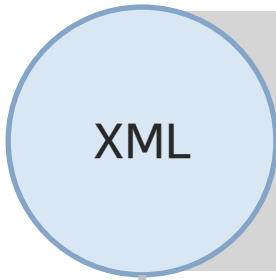
        // list 가져오기
        JSONArray msgList =(JSONArray)
jsonObject.get("list");
        Iterator<String>iterator = msgList.iterator();
        System.out.println("**JsonList**");
        while(iterator.hasNext()) {
            System.out.println(iterator.next());
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
}
}

```

1. 인터페이스 구현 도구

1 데이터 통신을 통한 인터페이스 구현

XML(eXtensible Markup Language)



다른 특수한 목적을 갖는 마크 업 언어를 만드는 데 사용하도록 권장하는
다목적 마크 업 언어

- 다른 많은 종류의 데이터를 기술하는 데 사용
- 다른 종류의 시스템 간 데이터를 쉽게 주고받는 것이 목적임

구분	설명
유니코드 문자	<ul style="list-style-type: none"> • 정의상 XML 문서는 문자로 이루어져 있음 • 한글과 같은 유니코드문자의 대부분은 XML문서로 표현 가능
XML 파서 (Parser)	<ul style="list-style-type: none"> • 파서는 마크 업을 분석하고 필요한 정보를 추출하여 애플리케이션에 넘김
마크 업 (Mark up)과 내용 (Content)	<ul style="list-style-type: none"> • XML 문서를 구성하는 문자들은 마크업과 내용으로 구분되고, 간단한 문법 규칙으로 이루어짐 • 마크 업으로 구성되는 문자열은 '<'로 시작하여 '>'로 끝나거나 '&'로 시작하여 문자 ';'로 끝나며 마크 업이 아닌 문자열은 내용임

1. 인터페이스 구현 도구

1 데이터 통신을 통한 인터페이스 구현

📦 XML(eXtensible Markup Language)

구분	설명
엘리먼트 (Element)	<ul style="list-style-type: none"> 문서의 논리 요소로서 시작 태그로 시작하여 짝이 되는 끝 태그로 끝나거나 빈 엘리먼트 태그만으로 이루어짐 자식 엘리먼트를 포함할 수 있음 <p>예 <code><Greeting><child>Hello world</child></Greeting></code></p>
어트리뷰트 (Attribute)	<ul style="list-style-type: none"> 엘리먼트 태그 속에 위치하는 속성 정보 <p>예 <code><step number="3">Connect A to B</step></code> number는 step 엘리먼트의 속성</p>
XML 선언	<ul style="list-style-type: none"> XML 문서는 다음과 같이 자신에 대한 정보 일부를 선언하는 것으로 시작 <p>예 <code><?xml version="1.0" encoding="UTF-8" ?></code></p>

1. 인터페이스 구현 도구

1 데이터 통신을 통한 인터페이스 구현

📦 XML 사용 예시

예 | XML을 사용하여 데이터 전송 포맷을 만든 예제

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<note>
<to>Tove</to>
<from>Jani</from>
<heading>Reminder</heading>
<body>Don't forget me this weekend!</body>
</note>
```

1. 인터페이스 구현 도구

2 인터페이스 엔티티를 통한 인터페이스 구현

◈ 인터페이스 엔티티를 통한 인터페이스 구현

인터페이스 엔티티를 통한
인터페이스 구현 방법이란?

인터페이스가 필요한 시스템 사이에
별도의 인터페이스 엔티티(데이터베이스)를 두어
각 시스템 간 데이터 교환에 활용하는 방법

1. 인터페이스 구현 도구

2 인터페이스 엔티티를 통한 인터페이스 구현

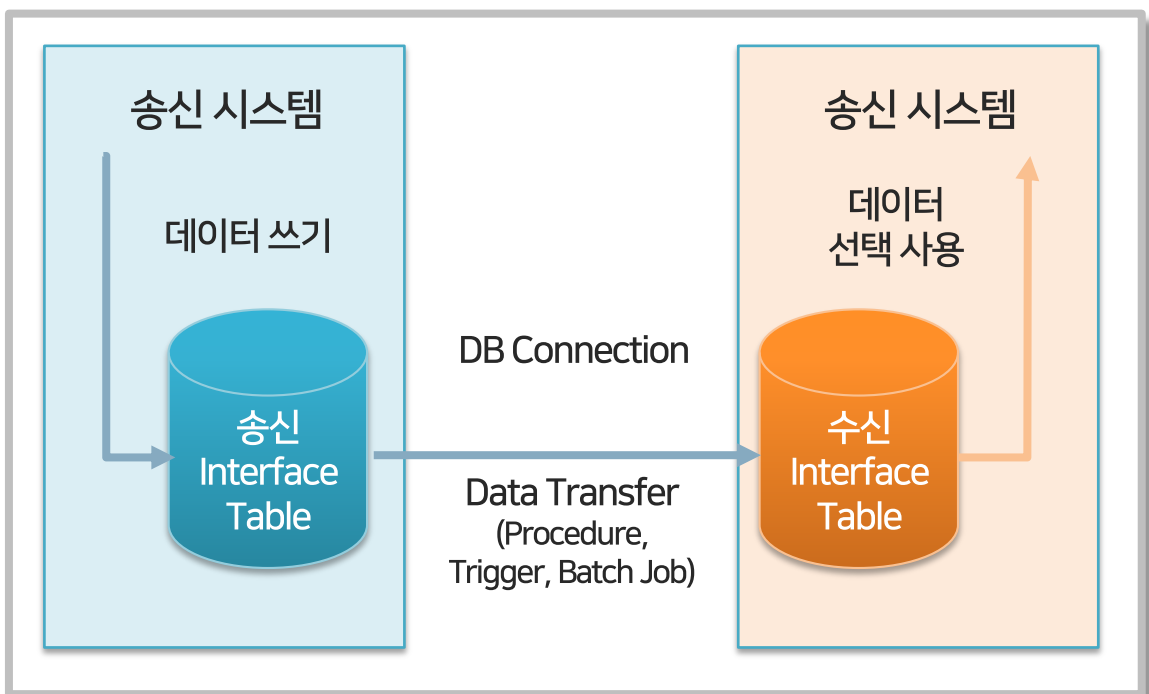
◆ 인터페이스 엔티티를 통한 인터페이스 구현

◆ 인터페이스 엔티티를 통한 인터페이스 구현 절차

송신 시스템에서 전달할 내역을
송신 인터페이스 테이블에 Write

송신 인터페이스 테이블에서 네트워크로 연결된 수신 인터페이스
테이블에 프로시저(Procedure), 트리거(Triiger),
배치 잡(Batch job) 등을 사용하여 데이터를 전달

수신 시스템은 전달된 데이터를 선택하여 활용



2. 인터페이스 구현 절차

1 기능 구현 분석

◆ 기능 분석 개요

기능 분석

사전에 상세히 정의된 기능 구현 내용을
어떻게 구현할 것인지 분석

- ◆ 개발하고자 하는 응용소프트웨어와 연계 대상 모듈 간의 세부 설계서를 확인하여 기능 구현을 정의하고 이를 분석하여 기능 구현에 활용
- ◆ 분석 시 구현 방법, 범위, 구체성 정도까지 고려하여 분석 후 빠르고 정확한 구현이 될 수 있도록 준비

2. 인터페이스 구현 절차

1 기능 구현 분석

◆ 기능 분석 개요

예 | 기능 구현 분석

기능 구현 정의	상세 기능	상세 기능
1. 송신 측에서 필요 대상을 선택하여 관계사에 데이터 전송	(1) 대상자 선택	DB에 저장된 인사 정보를 SQL로 선택하여 인터페이스 대상 확인
	(2) 인터페이스 데이터 생성	JSON 또는 인터페이스 테이블 형식에 맞게 DB에 있는 정보 가공
	(3) 인터페이스 데이터 전송 요청	인터페이스 데이터를 생성 후 해당 데이터를 http에서 인터페이스 또는 수신 측으로 전송

2. 인터페이스 구현 절차

1 기능 구현 분석

◆ 기능 분석 개요

예 | 기능 구현 분석

기능 구현 정의	상세 기능	상세 기능
2. 필요 항목에 대해서 데이터 전달	(1) 인터페이스 데이터 전송	수신지 주소에 <ul style="list-style-type: none"> • REST 방식으로 전달 • DB Connection을 이용하여 DB Procedure, Trigger 등을 활용, 전송
	(2) 인터페이스 데이터 수신	<ul style="list-style-type: none"> • 인터페이스 데이터를 수신하고 데이터의 형식이 맞는지 점검 • 각 Rule을 사전에 정의하고 활용
	(3) 인터페이스 데이터 파싱	인터페이스 객체를 파서를 이용하여 파싱하거나 인터페이스 테이블에서 SELECT 함
	(4) 인터페이스 데이터 검증	각 세부 데이터의 포맷과 논리적인 제약조건에 걸리지 않는지 검사 <div> 예 사용자가 정의한 룰: salary는 0원보다 많아야 함 </div>
	(5) 후속 기능 수행	각 세부 데이터를 활용해서 수신 시스템에서 정의된 Action 진행

2. 인터페이스 구현 절차

1 기능 구현 분석

📌 기능 분석 개요

예 | 기능 구현 분석

기능 구현 정의	상세 기능	상세 기능
3. 전달 후 전달받은 관계사로 부터 전달 여부 값을 반환 받음	(1) 인터페이스 수신 결과를 생성, 전송	정상적으로 신뢰되면 True, 아니면 False의 Return 값을 JSON 형태로 전송
	(2) 수신 측에서 처리한 인터페이스 결과를 반환 받음	수신 측에서 보낸 응답 JSON을 통해 수행 여부를 확인하고 이를 화면상으로 표시

2. 인터페이스 구현 절차

2 기능 구현 정의

❖ 데이터 통신을 사용하는 인터페이스 구현

- ❖ 데이터 통신을 사용하는 인터페이스 구현 사례 중 최근 경량화 되고 가장 많이 사용하는 JSON을 이용한 인터페이스 구현 사례를 다루어 봄



2. 인터페이스 구현 절차

2 기능 구현 정의

◆ 데이터 통신을 사용하는 인터페이스 구현

인터페이스 구현 절차

1 '인터페이스 객체 생성'까지 구현

- 인터페이스 객체를 생성하고 전송하기 위해 인터페이스 객체를 생성할 데이터 선택
- 선택한 데이터는 JSON을 이용하여 인터페이스 객체를 만드는 데 사용

- 인터페이스 객체를 생성할 데이터를 각 시스템 및 환경에 맞게 선택
- 일반적으로 데이터베이스에 있는 정보를 SQL을 통하여 선택한 후 이를 JSON으로 생성

2. 인터페이스 구현 절차

2 기능 구현 정의

❖ 데이터 통신을 사용하는 인터페이스 구현

인터페이스 구현 절차

1 '인터페이스 객체 생성'까지 구현

① 인터페이스 객체를 생성하기 위한 데이터 선택

```
Statement stmt; // sql을 실행하는 자료 구조
ResultSet rs; // 실행한 sql을 저장하는 자료 구조
String sql; sql을 작성할 변수
sql = " SELECT EMPNO, NAME, JIKGUB, ORG_NM ..
        \n FROM EMP_HISTORY
        \n WHERE EMPNO = 'XXX' --사번
        \n AND HISTORY_NO = 'DDD' -- 발령 번호"
try{
    rs = stmt.executeQuery(sql); // sql 실행 rs에
저장
}catch(Exception e){
    throw e;
}
return rs; //실행 결과 return
```

2. 인터페이스 구현 절차

2 기능 구현 정의

◆ 데이터 통신을 사용하는 인터페이스 구현

인터페이스 구현 절차

1 '인터페이스 객체 생성'까지 구현

② 인터페이스 객체 구현

- SQL 실행 결과를 저장한 객체를 화면으로 가져와서 JSON 객체 생성
- JSON 형식에 맞게 JavaScript 객체에 저장

```
JSONObject obj = new JSONObject();  
obj.put("EMONP",{rs.EMPNO}); //jquery로 데이터를  
매핑  
obj.put("NAME",{rs.NAME});  
....
```


2. 인터페이스 구현 절차

2 기능 구현 정의

◆ 데이터 통신을 사용하는 인터페이스 구현

인터페이스 구현 절차

2 인터페이스 객체를 전송 후 전송 결과를 수신 측에서 반환

- 작성된 인터페이스 객체는 수신 측으로 데이터 통신을 통하여 전송
- 해당 결과를 수신 측에서 송신 측으로 전송하면 최종 인터페이스 여부를 판단

2. 인터페이스 구현 절차

2 기능 구현 정의

◆ 데이터 통신을 사용하는 인터페이스 구현

인터페이스 구현 절차

2 인터페이스 객체를 전송 후 전송 결과를 수신 측에서 반환

① 송신 측에서 인터페이스 객체를 수신 측에 송신

- JSON으로 작성된 객체를 수신 측에 송신
- 대부분 AJAX 기술을 이용하여 JSON을 수신 측에 전달

```
JSONObject json =
(JSONObject)JSONValue.parse(jsonParam);
..... // json 세팅
$.ajax({
  type : "POST",
  data Type : "json"
  url : "/company/receiveInterface" // 수신 받는 주소
  data : {
    json : obj
  },
}) // click 같은 이벤트로 데이터 전달 시에 이벤트 function
// 안에 해당 내용을 포함한다.
```

2. 인터페이스 구현 절차

2 기능 구현 정의

◆ 데이터 통신을 사용하는 인터페이스 구현

인터페이스 구현 절차

2 인터페이스 객체를 전송 후 전송 결과를 수신 측에서 반환

① 송신 측에서 인터페이스 객체를 수신 측에 송신

- 수신 측에서는 JSON 인터페이스 객체를 수신하여 파싱 후 처리
- 처리 결과 값을 송신 측에 전달
- 일반적으로 True/False를 전달하여 인터페이스 성공 여부 확인

```
JSONObject json =  
(JSONObject)JSONValue.parse(jsonParam);  
/* 인터페이스 객체에서 데이터를 select */  
String if_name = (String)json.get("NAME")  
  
..... // 데이터를 select하고 이를 트랜잭션 한다.  
  
response.setStatus(200) // success를 나타냄
```

2. 인터페이스 구현 절차

2 기능 구현 정의

❖ 인터페이스 엔티티를 사용한 인터페이스 구현

- ❖ 독립적인 인터페이스 엔티티를 사용하여 인터페이스를 구현하는 경우 일반적으로 인터페이스 테이블을 엔티티로 활용
- ❖ 1개의 인터페이스 테이블을 사용할 수도 있고 송신 및 수신 인터페이스 테이블을 각각 사용할 수도 있음
- ❖ 송신 테이블과 수신 테이블의 구조는 같은 경우가 대부분이나 경우에 따라서는 다르게 설계할 수도 있음

2. 인터페이스 구현 절차

2 기능 구현 정의

◆ 인터페이스 엔티티를 사용한 인터페이스 구현

인터페이스 구현 절차

1 송신 측 인터페이스 테이블에 데이터를 Write하고 전송

- 송신 시스템의 인터페이스 테이블에서는 송신 관련 정보(연계 데이터 생성 일시, 송신 일시, 송신 상태 등)를 관리하기 위한 항목과 송신 시스템에서 필요한 항목을 추가하여 설계

① 송신 측에서 테이블에 데이터를 Write

- 인터페이스 이벤트 발생 시에 인터페이스 테이블에 인터페이스 내용을 Write함
- Write 동작은 내용의 신규 추가만 가능하며 수정과 삭제 처리는 할 수 없음 (데이터 무결성 유지 및 인터페이스 이력 관리를 위함)
- Write 기능과 같은 처리를 위하여 테이블 생성 시 제약조건문장(Privilege 등)을 기입하여 처리할 수 있음

2. 인터페이스 구현 절차

2 기능 구현 정의

◆ 인터페이스 엔티티를 사용한 인터페이스 구현

인터페이스 구현 절차

1 송신 측 인터페이스 테이블에 데이터를 Write하고 전송

② 송신 측에서 테이블에 데이터를 전송

- 정의된 인터페이스 시점(즉시 또는 특정 기간 이후, 주기적)에 수신 측으로 데이터를 전송할 수 있음
- 전송하기 위해서는 일반적으로 DB Connection이 수신 측 인터페이스 테이블과 맺어져 있어야 함
- Procedure, Trigger, Batch Job 등의 방법을 통해서 수신 테이블로 전송

2. 인터페이스 구현 절차

2 기능 구현 정의

◆ 인터페이스 엔티티를 사용한 인터페이스 구현

인터페이스 구현 절차

2 수신 측 인터페이스 테이블에서 인터페이스 데이터를 Read 하고 처리

- 수신 시스템의 인터페이스 테이블에서는 수신 관련 정보(수신 일시, 처리 일시, 수신 상태 등)를 관리하기 위한 항목과 수신 시스템에서 필요한 항목을 추가하여 설계

① 수신 측 인터페이스 테이블에서 인터페이스 데이터를 Read함

- 인터페이스 데이터가 수신 측 인터페이스 테이블에 입력되면 정해진 주기(즉시 또는 주기적)에 따라 데이터를 읽음

2. 인터페이스 구현 절차

2 기능 구현 정의

◆ 인터페이스 엔티티를 사용한 인터페이스 구현

인터페이스 구현 절차

2 수신 측 인터페이스 테이블에서 인터페이스 데이터를 Read 하고 처리

① 수신 측 인터페이스 테이블에서 인터페이스 데이터를 Read함

구분	예외 원인	예외 처리 방안
수신 일시	데이터 수신 일시	
변경 구분	입력, 수정, 삭제	운영 DB 처리 시 참고
연계 데이터 1	송신 시스템에서 송신한 데이터	운영 DB 반영 사항
연계 데이터 2		
.....		
연계 데이터 N		
송신 일시	데이터 송신 일시	운영 DB 처리 순서 구분
처리 일시	수신 후 처리 일시	
처리 상태	성공 / 실패 / 보류	
오류 코드	오류 발생 시 코드	
오류 내용	오류 발생 상세 내용	

2. 인터페이스 구현 절차

2 기능 구현 정의

◆ 인터페이스 엔티티를 사용한 인터페이스 구현

인터페이스 구현 절차

2 수신 측 인터페이스 테이블에서 인터페이스 데이터를 Read 하고 처리

① 수신 측 인터페이스 테이블에서 인터페이스 데이터를 Read함

- 수신 측 시스템에서는 인터페이스 데이터를 읽은 후 사전에 정의된 데이터 트랜잭션 진행
- 데이터를 읽거나 해당 트랜잭션이 진행될 때 오류가 발생하면 오류 코드 칼럼에 정의된 오류 코드 및 오류 내용 입력
- 인터페이스 테이블 활용 시 DB에 부하를 줄 수 있는 **트리거를 이용한 데이터 전송 지양**
- JSON 생성을 위해 SQL Read 및 데이터 Write할 때 각자 시스템 환경 및 개발 도구에 맞게 적절한 기법 사용

핵심요약

인터페이스 구현 도구

■ 인터페이스를 구현하는 대표적인 방법

- 데이터 통신을 이용한 인터페이스 구현 방법
- 인터페이스 테이블을 이용한 인터페이스 구현 방법

■ 데이터 통신을 이용한 인터페이스 구현 방법

애플리케이션 영역에서 인터페이스 형식에 맞춘 데이터 포맷을 인터페이스 대상으로 전송하고 이를 수신 측에서 파싱(Parsing)하여 해석하는 방식으로 주로 JSON 및 XML 형식의 데이터 포맷을 사용하여 인터페이스를 구현함

■ 인터페이스 엔티티를 통한 인터페이스 구현 방법

데이터베이스에서 인터페이스 테이블을 두어 각 시스템 간 데이터 교환에 활용하는 방식으로 여러 업계에서 많이 사용됨

핵심요약

인터페이스 구현 절차

■ 인터페이스 구현 절차

- ① 도출된 인터페이스 기능 분석
- ② 데이터 통신 또는 인터페이스 엔티티를 사용하여
인터페이스 구현