공통 모듈 테스트 및 예외 처리 방안

한국기술교육대학교
온라인평생교육원

학습내용

- 공통 모듈 테스트
- 공통 모듈 예외 처리 절차

학습목표

- 공통 모듈 테스트 절차를 설명할 수 있다.
- 공통 모듈 테스트를 위한 테스트 케이스를 설계하고, 테스트 계획을 수립할 수 있다.
- 공통 모듈 테스트 수행 시 발생할 수 있는 예외의 처리 절차를 설명할 수 있다.

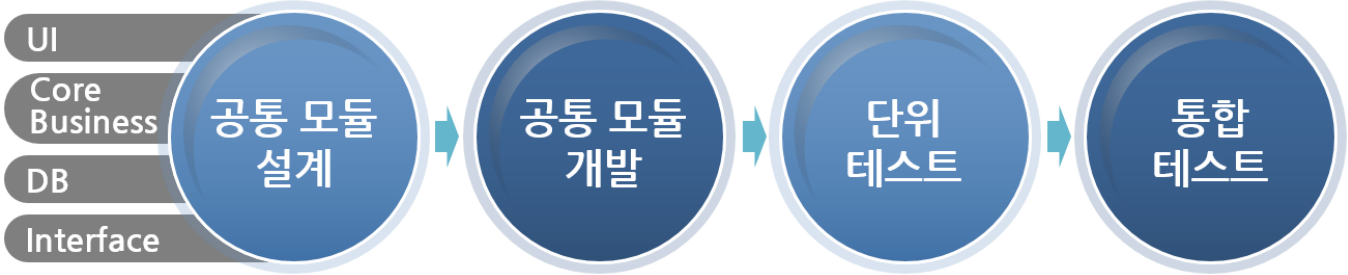
공통 모듈 테스트

공통 모듈 테스트의 개요

공통 모듈 유형

UI 영역	Core Business 영역	DB 영역	Interface 영역
• 권한처리 • 파일첨부 • 마스킹 • 파라메터 암호화 • 공통 코드 • 로그 저장 • 시스템 메시지 처리 • UI Session Timeout	• Single Sign On • 암복호화 • 거래제어 • 배포 연계 • 예외 처리 • 배치 자동화 • SOAP Header • 시스템 유틸리티 • 형상관리	• DB Connection • DB Session 관리 • DB Transaction 관리 • 프로시저, 함수 관리 • SQL Binding	• Interface DB 로깅 • 연동 솔루션 모니터링 및 통계 • 배포 자동화 • 파라메터 암호화 • 전문 암/복호화 • 예외 처리 및 재처리

공통 모듈 설계 및 테스트 절차

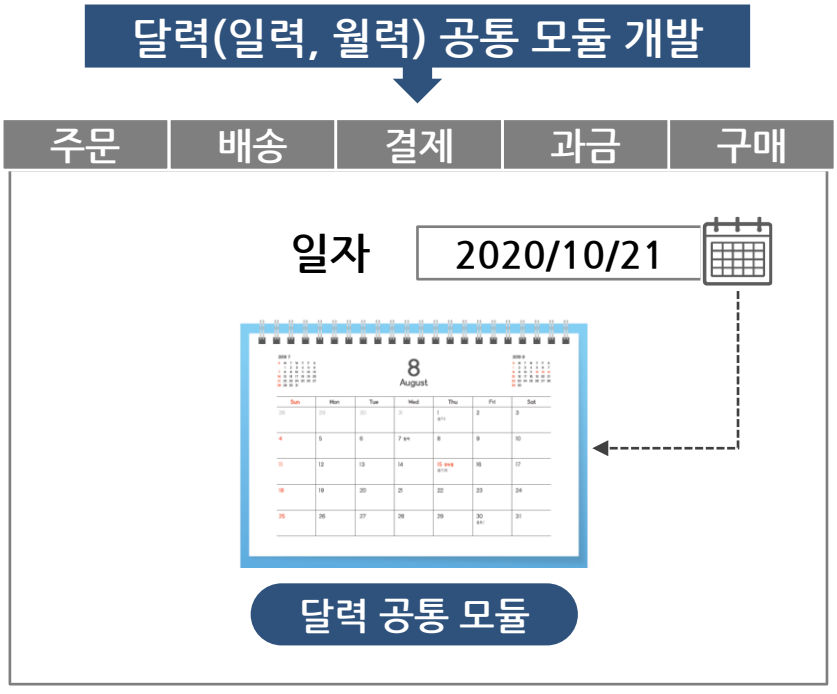


공통 모듈 테스트

공통 모듈 단위 테스트

수행 내용

- 공통 모듈 개발 후 자체 단위 테스트 수행
  (해당 공통 모듈에 대해서만 테스트)
- 필요한 경우 스텁(Stub) 혹은 드라이버(Driver)를 만들어 테스트 수행
  - 스텁 : 하위 모듈 개발이 되지 않았을 경우 임시로 만드는 테스트 모듈
  - 드라이버 : 상위 모듈 개발이 되지 않았을 경우 임시로 만드는 테스트 모듈



공통 모듈 테스트

공통 모듈 단위 테스트

공통 모듈 단위 테스트 시나리오 도출

UI Session timeout 기능 명세

영역 구분 (공통 모듈명)	함수명	함수 설명

FRM_TOP.xfdl	this.top_ontimer()	타이머 적용
		case 20200519 : // 토큰 유효성 체크 서비스 호출
		this.fn_extndUserAthnTkn();
		case 999 : // 1시간 동안 서비스 없는지 체크
		var popupCmfr = this.fn_UseChk();

FRM_TOP.xfdl	this.fn_UseChk()	사용여부 확인 (57분 동안 사용이 없을 경우를 확인)
FRM_TOP.xfdl	this.fn_extndUserAthnTkn()	포털세션 끊김 현상 개선을 위한 인증토큰 연장 Process

공통 모듈 테스트

공통 모듈 단위 테스트

공통 모듈 단위 테스트 시나리오 도출

UI Session timeout 단위 테스트 시나리오

테스트 명	테스트 시나리오
세션 연장 처리 테스트	• 세션 체크하여 온라인 운영 시간 10분에 한번씩 체크하는지 테스트 • 세션 체크 오류 시 UI 강제 종료함
종료 확인 창 테스트	• 57분 동안 서비스 호출이 발생하지 않는 경우, 종료 확인 창을 사용자에게 보여주는지 테스트 • 3분 이내 미확인 시, 자동으로 UI 종료 처리함

공통 모듈 테스트

공통 모듈 단위 테스트

공통 모듈 단위 테스트 시나리오 도출

정합성 체크 기능 명세

영역 구분 (공통 모듈명)	함수명	함수 설명
LIB_VALIDATION.xjs	this.gfn_isValidJuminNo =function(strJuminNo, bMsgLock)	주민등록번호 적합성 여부 체크 함수
LIB_VALIDATION.xjs	this.gfn_isValidForeignNo =function(strForeignNo, bMsgLock)	외국인 등록번호 적합성 여부 체크 함수
LIB_VALIDATION.xjs	this.gfn_isValidBizNo =function(strBizNo, bMsgLock)	사업자 등록번호 적합성 여부 체크 함수

공통 모듈 테스트

공통 모듈 단위 테스트

공통 모듈 단위 테스트 시나리오 도출

정합성 체크 단위 테스트 시나리오

테스트 명	테스트 시나리오
주민등록번호 정합성 테스트	사용되는 주민번호 입력, 가상의 주민번호 입력
외국인등록번호 정합성 테스트	사용되는 외국인등록번호 입력, 가상의 외국인등록번호 입력
사업자등록번호 정합성 테스트	사용되는 사업자등록번호 입력, 가상의 사업자등록번호 입력

공통 모듈 테스트

공통 모듈 통합 테스트

- 애플리케이션 통합 테스트 시나리오에 따라 UI, Core Business, DB, Interface 공통 모듈을 한꺼번에 테스트

애플리케이션 통합 테스트 시나리오

UI 공통 모듈	Core Business 공통 모듈	DB 공통 모듈	Interface 공통 모듈
• 파일첨부 • 마스킹 • 달력 Popup • 공통 코드 • 로그 저장 • UI Session Timeout	• Single Sign On • 암복호화 • 거래 제어 • 배포 연계 • 예외 처리	• DB Connection • DB Session 관리 • DB Transaction 관리 • 프로시저, 함수 관리 • SQL Binding	• Interface DB 로깅 • 연동 솔루션 모니터링 및 통계 • 배포 자동화 • 파라메터 암호화 • 전문 암/복호화 • 예외 처리 및 재처리

공통 모듈 예외 처리 절차

예외 처리 프로세스 설계

- 공통 모듈 예외 처리를 위한 프로세스 설계
- 데이터베이스 Rollback 처리
- 화면 잠금 혹은 Close 처리

예외 처리 공통 메시지 설계

- 예외 처리를 위한 공통 코드 설계
- 메시지 타입 설계

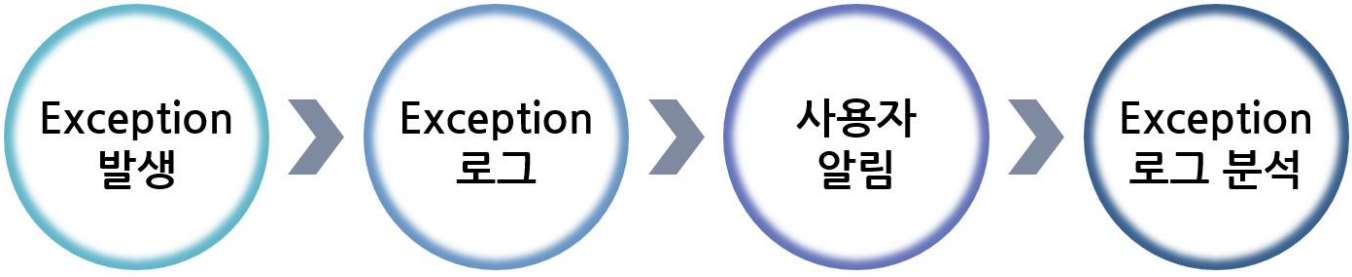
로그 설계

- 오류 로그 기록 설계
- 로그 레벨 정의

공통 모듈 예외 처리 절차

예외 처리 프로세스 설계

절차



예외 유형

구분	설명	구현 영역
솔루션 Exception	• 미들웨어, DB 등 솔루션에서 발생시키는 Exception	• 솔루션 Engine에서 사용
비즈니스 Exception	• 미들웨어, DB 등 솔루션을 이용하여 Business 업무 구현 영역에서 발생하는 Exception • 개발자가 업무적인 오류 발생 시 사용하는 Exception객체	• 개발자가 업무 구현 시 오류 유형을 Catch하기 위하여 사용
기타 Exception	• 일반적으로 모든 Exception을 포함하는 객체 • System 상에서 발생하는 오류 유형	• 시스템 영역에서 예외 발생 시 생성됨

공통 모듈 예외 처리 절차

공통 모듈 예외 처리를 위한 메시지 타입 설계

메시지 코드 포맷

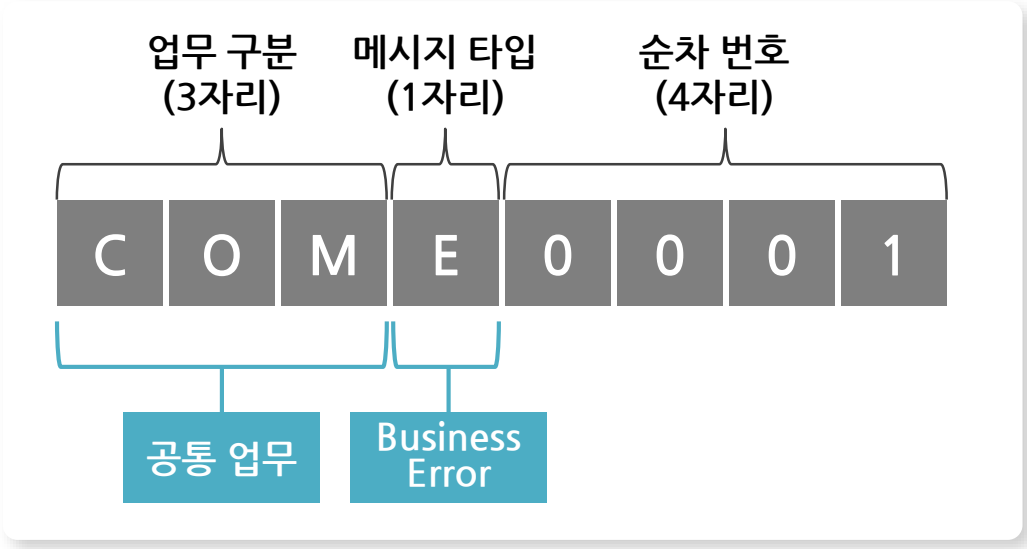
- 공통 코드로 미리 정해 놓고 전체 프로젝트에서 사용

유형	응답 메시지 구분		활용 목적
S	System Error	에러 메시지	• 서비스 수행결과가 시스템 오류인 경우에 사용하는 메시지
E	Business Error		• 서비스 수행결과가 업무 오류인 경우에 사용되거나 미입력 또는 잘못된 값 등으로 인해 발생하는 오류에 사용되는 메시지
I	Information	정상 메시지	• 서비스 수행결과가 성공인 경우 • 정상 처리 메시지를 필요 시 전달 • UI Popup 대상 아님
D	Dialog		• 서비스 수행결과가 성공이나 추가적인 전달 내용이 있는 경우

공통 모듈 예외 처리 절차

공통 모듈 예외 처리를 위한 메시지 타입 설계

메시지 코드 포맷 예



공통 모듈 예외 처리 절차

공통 모듈 예외 처리 메시지 반환 로직 설계

```
if (e instanceof ProObjectException)
      e = (Exception) e.getCause();

if (e instanceof NullPointerException) {
      return “COME0001”;
} else if (e instanceof SQLException) {
     switch (e.getErrorCode()) {
       case 1:   // ORA-00001 (무결성 제약 조건에 위배됩니다.)
                 return “COME0001”;
       case 984 : // ORA-00984 (열을 사용할 수 없습니다.)
                 return “COME0002”;
       case 1400 :
                 return “COME0003”;
       case 1722 :
                 return “COME0004”;
       case 3114 :
                 return “COME0005”;
     }
}
```

공통 모듈 예외 발생 시 최초 발생한 Exception을 추적하여 오류 유형 판단 가능

공통 모듈 예외 처리 절차

공통 모듈 예외 로그 확인

서버 로그

- 서버에서 수행되는 공통 모듈의 수행 결과는 서버 로그를 통해 확인
- 서버 로그는 서버마다 다른 디렉토리에서 저장됨

경로	출력 내용
/applog/PRO_OBJECT/server1/proobject.log	ProObject 프레임워크가 수행하는 전문 처리 등의 로그
/applog/PRO_OBJECT/server1/prologs/ProObject_yyyymmdd.log	Business Logic 처리 로그 및 관련 프레임워크 로그
/applog/PRO_OBJECT/server1/prologs/{패키지경로}/{SO명 혹은 JO명}_yyyymmdd.log	SO(Service Object) 또는 JO(Job Object) 단위의 Business Logic 처리 로그 및 관련 프레임워크 로그

공통 모듈 예외 처리 절차

공통 모듈 예외 로그 확인

tail 명령어

- 리눅스 tail 명령어를 이용하여 실시간 로그 확인
- tail [option]….[file]…..
  예 tail -n 20 anaconda-ks.cfg

```
loginluv@ip-172-31-13-54:~$ tail -f /var/log/dmesg
[    4.372158] type=1400 audit(1418392061.833:7): apparmor="STATUS" operation="profile_replace" pro
file="unconfined" name="/usr/lib/connman/scripts/dhclient-script" pid=488 comm="apparmor_parser"
[    4.789173] random: nonblocking pool is initialized
[    6.489076] init: failsafe main process (739) killed by TERM signal
[    6.712687] type=1400 audit(1418392064.173:8): apparmor="STATUS" operation="profile_replace" pro
file="unconfined" name="/sbin/dhclient" pid=854 comm="apparmor_parser"
[    6.712693] type=1400 audit(1418392064.173:9): apparmor="STATUS" operation="profile_replace" pro
file="unconfined" name="/usr/lib/NetworkManager/nm-dhcp-client.action" pid=854 comm="apparmor_parse
r"
[    6.712697] type=1400 audit(1418392064.173:10): apparmor="STATUS" operation="profile_replace" pr
ofile="unconfined" name="/usr/lib/connman/scripts/dhclient-script" pid=854 comm="apparmor_parser"
[    6.713008] type=1400 audit(1418392064.173:11): apparmor="STATUS" operation="profile_replace" pr
ofile="unconfined" name="/usr/lib/NetworkManager/nm-dhcp-client.action" pid=854 comm="apparmor_pars
er"
[    6.713011] type=1400 audit(1418392064.173:12): apparmor="STATUS" operation="profile_replace" pr
ofile="unconfined" name="/usr/lib/connman/scripts/dhclient-script" pid=854 comm="apparmor_parser"
[    6.713167] type=1400 audit(1418392064.173:13): apparmor="STATUS" operation="profile_replace" pr
ofile="unconfined" name="/usr/lib/connman/scripts/dhclient-script" pid=854 comm="apparmor_parser"
[    6.727908] type=1400 audit(1418392064.185:14): apparmor="STATUS" operation="profile_load" profi
le="unconfined" name="/usr/sbin/tcpdump" pid=856 comm="apparmor_parser"
^C
```

학습정리

1. 공통 모듈 테스트

- 공통 모듈은 UI, Core Business, DB, Interface 영역 등에 걸쳐서 다양하게 존재함
- UI, Core Business, DB, Interface 등 각 영역에 걸쳐서 공통 모듈 개발 후 단위 테스트와 통합 테스트를 진행함
- 공통 모듈 개발 후 자체 단위 테스트를 수행함
  - 필요한 경우 스텁(Stub) 혹은 드라이버(Driver)를 만들어 테스트함
- 애플리케이션 통합 테스트 시나리오에 따라 UI, Core Business, DB, Interface 등 공통 모듈을 한꺼번에 테스트함

2. 공통 모듈 예외 처리 절차

- 공통 모듈 예외 처리를 위해서 프로세스 설계, 메시지 및 로그 설계를 함
- 공통 모듈 예외가 발생하면 시스템에 로그 기록부터 하고, 사용자에게 예외 상황을 알린 후 로그 분석을 함
- 공통 모듈 예외 유형
  - 비즈니스 로직 상 개발자가 발생시키는 Exception
  - 미들웨어, DB 등 솔루션에서 발생시키는 Exception
  - 그 외 시스템에서 발생시키는 모든 Exception