[별첨]

Zookeeper 테스트 케이스

2013. 05

nipa 정보통신산업진흥원 National IT Industry Promotion Agency
nipa 공개SW역량프라자 Open Source Software COMPENTENCE PLAZA

Stack A,B,C 공통

순번	대분류	중분류	시나리오명	시나리오 개요	시나리오 흐름	케이스 번호	케이스	입력데이터	예상결과	결과	오류증상	비고
1	시작/종료		Zookeeper 실행/종료	Zookeeper을 시작한 후 종료한다.	1.Zookeeper서버를 시작한다.	1	명령 프롬프트에서 명령어를 입력한다.	zkServer.sh start	명령 프롬프트가 실행된다.	pass		
						2	Zookeeper을 시작한다.					
					2.Zookeeper서버를 종료한다.	1	Zookeeper을 종료한다.	zkServer.sh stop	flume이 종료된다	pass		
2	Monitering		모니터링	기본적인 정보를 모니터 한다.	모니터링을 시작한다.	1	Zookeeper Eclipse Plug-in을 설치한다.		Zookeeper 정보가 확인된다.	pass		
3	Zookeeper Client API	create	트리의 위치에 노드를 생성	특정 노드를 생성한다.	특정 노드를 생성한다.	1	Zookeeper 객체를 생성한다.	Zookeeper zk=new Zookeeper();	주키퍼 객체를 생성한다.	pass		
						2	create문으로 특정노드 생성	zk.create("/test01","test01_data".getbyte().Ids.OPEN_ACL_UNSAFE,CreateMode.PERSISTENT);	부모 혹은 자식노드가 생성된다.	pass		
		delete	노드를 삭제	특정 노드를 삭제한다.	특정 노드를 삭제한다.	1	delete 문으로 특정 노드 삭제	zk.delete("/test01");	부모 혹은 자식노드를 삭제한다.	pass		
		exists	노드가 존재 하는지 테스	특정 노드 확인	특정 노드 확인	1	exists문으로 특정노드가 있는지 확인한다.	(zk.exists("/test01",false)!=null);	부모 혹은 자식노드를 확인한다.	pass		
		getData	노드의 데이터 읽어옴	특정 노드의 데이터를 읽어온다.	특정 노드의 데이터를 읽어온다	1	getData분으로 특정노드의 데이터를 가져온다	byte[] test01Data = zk.getdata("/test01",false,null);	노드의 데이터를 읽어온다.	pass		
		setData	노드의 데이터를 기록	특정 노드의 데이터를 기록한다.	특정 노드의 데이터를 기록한다	1	setData문으로 특정 노드의 데이터를 저장한다	zk.setData("/test01","this is new data".getByte(),-1);	노드에 데이터를 기록한다.	pass		
		getChildren	노드의 자식 노드 리스트를 검색	특정 노드의 자식 리스트를 가져온다	특정 노드의 자식 리스트를 가져온다	1	리스트로 자식노드 목록을 가져온다	List<String> children=zk.getChildren("test01",false);	노드의 자식 노드 리스트를 가져온다.	pass		
		close	클라이언트 접속을 종료한다	클라이언트 접속을 종료한다.	클라이언트 접속을 종료한다.	1	주키퍼 클라이언트 종료	zk.close()	주키퍼 객체를 종료한다.	pass		
4	분산락 구현		분산락 구현	Zookeeper API를 이용하여 분산락을 구현한다.	Zookeeper API를 이용하여 분산 락을 구현한다.	1	부모노드 생성		노드 lock 생성	pass		
						2	클라이언트 1에서 lock요청		순번 배정이 이루어짐 01	pass		
						3	클라이언트 1에서 lock받아옴		01 프로세스 업무 시작	pass		
						4	클라이언트 2에서 lock요청		순번 배정이 이루어짐 02	pass		
						5	클라이언트 2에서 lock받아옴		01이 작업중이므로 대기	pass		
						6	클라이언트 1에서 lock삭제		01 프로세스 종료/ 모든 클라이언트에 이벤트 발생	pass		
						7	클라이언트 2에서 lock받아옴		기존의 lock정보와 비교 02=02 면 업무 처리 02!=02면 종료	pass		
						8	클라이언트 2에서 lock 삭제		02 프로세스 종료/모든 클라이언트에 이벤트 발생	pass		
5	로드밸런싱		로드밸런싱 구현	Zookeeper API를 이용하여 로드밸런싱을 구현한다.	Zookeeper API를 이용하여 로드밸런싱을 구현한다.	1	WAS서버 시작			pass		
						2	서버 노드가 없으면 생성	/server	서버 노드가 생성됨	pass		
						3	ip,port 번호로 서버 정보 생성(3대의 서버 등록)	121.162.249.18:8080, 121.162.249.19:8080 ,121.162.249.23:8080	서버의 자식노드3개 등록	pass		
						4	사용할 서버 정보 노드 생성		자식노드의 데이터에 서버정보 기록	pass		
						5	클라이언트 접속 1번서버에 접속		클라이언트에서 메인서버에 접속 정보 요청	pass		
						6	1번(121.162.249.18) 서버 장애 발생(이벤트 발생)		메인서버에서 서버1의 세션 및 노드 정보를 삭제하고 모든 접속한 클라이언트에 이벤트 전달	pass		
						7	클라이언트에서 메인서버로 정보요청		클라이언트는 메인서버에 새로운 접속정보를 요청	pass		
						8	메인서버에서 정상적인 서버정보를 전달		정상적인 서버정보(2번서버)를 클라이언트에 전달 (121.162.249.19:8080)	pass		
6	세션	CONNECTING	세션이 연결된 상태	Zookeeper API를 이용하여 세션 연결을 구현한다.	Zookeeper API를 이용하여 세션 연결을 구현한다.	1	클라이언트와 주키퍼서버 중 하나와 연결을 시도하고 있는 상태		연결 시도중 메시지 발생	pass		
		CONNECTED	세션이 연결된 상태에서 z노드에 연산 수행			2	클라이언트가 서버 중 하나와 연결된 상태로 z노드에 연산을 수행 할 수 있다		z노드에 연산을 수행 할 수 있음	pass		
		CLOSED	close() 메소드 호출에 의해 명시적으로 세션이 종료된 상태			3	close() 메소드 호출에 의해 명시적으로 연결을 종료했거나 세션타임 아웃,인증 실패 등의 이유로 세션이 종료된상태		세션과 연결 모두 종료된상태	pass		
		DISCONNECTED	특정서버와 연결이 끊긴 상태			4	특정 서버와 접속이 끊긴 상태, 주키퍼 서버가 여러대 이면 Zookeeper객체를 생성할 때 서버 정보를 넘겨준다		세션은 종료되고 클라이언트가 생성한 임시노드는 삭제 되지 않는다.	pass		
		SESSION EXPIRED	클라이언트가 주키퍼 서버로 접속 할 때 파라메터로 세션 타임 아웃 값을 전달			5	클라이언트가 주키퍼 서버로 접속 할 때 파라미터로 세션타임아웃 값을 전달 한다.		세션 재인증 및 클라이언트가 생성한 임시노드 삭제	pass		