
마이크로서비스 시나리오

목차

1. Socket.....	2
2. TCP/IP Socket Programming	2
2.1 Client Socket 과 Server Socket.....	2
2.2 Socket API 실행 흐름	3
3. Socket Error.....	3

1. Socket

네트워크 환경에 연결할 수 있게 만들어진 연결부로 OSI 7 layer에서 TCP layer상에서 동작하는 소켓을 주로 사용하며 TCP/IP 소켓이라고 부른다.

2. TCP/IP Socket Programming

소켓을 사용하여 네트워크 통신 기능을 구현하는 과정으로 운영체제 및 개발 언어에 종속적으로 제공되는 소켓 API사용법을 숙지해야한다.

추가로 네트워크 단절, 트래픽 증가에 따른 데이터 전송 지연, 시스템 리소스 관리문제로 인한 에러 등, 네트워크 환경에서 발생할 수 있는 다양한 예외사항에 대해 고려해야한다.

2.1 Client Socket과 Server Socket

두 개의 시스템이 소켓을 통해 네트워크 연결을 만들기 위해서는, 최초 어느 한곳에서 그 대상이 되는 곳으로 연결을 요청해야한다. IP 주소와 포트 번호로 식별되며 네트워크 연결 가능 상태임을 알린다.



요청을 받는 곳에서는 어떤 연결 요청을 받아들일 것인지 미리 시스템에 등록하여, 요청이 수신되었을 때 해당 요청을 처리할 수 있도록 준비해야 한다.

두 개의 시스템이 소켓을 통해 데이터 통신을 위한 연결을 만들기 위해서는 연결 요청을 보내는지 또는 요청을 받는지에 따른 역할을 나누게 되는데 전자의 연결을 요청하는 소켓은 Client Socket, 후자는 Server Socket으로 불린다.



위 두 소켓은 역할만이 다르고 소켓의 형태는 동일하며, 호출되는 API 함수의 종류와 순서들이 다르다는 차이만이 있다. 실제로 데이터를 주고받는 소켓은 추가로 새로운 소켓을 통해 처리되며, 위 두 소켓은 오로지 connection을 위한 소켓이다.

2.2 Socket API 실행 흐름

Client Socket의 실행 흐름

1. Socket을 생성한다(create)
2. Server socket에 연결을 요청한다(connect)
3. 연결이 승인되면 데이터를 송수신한다(send/rcv)
4. 모든 처리가 완료되면 Socket을 닫는다(close)

Server socket의 실행 흐름

1. Socket을 생성한다(create)
2. 서버가 사용할 IP주소와 포트 번호를 생성한 소켓에 결합시킨다(Bind)
3. 연결 요청이 수신되는지 확인한다(listen)
4. 요청이 수신되면 요청을 받아들여 데이터 통신을 위한 Socket을 생성한다(accept)
5. 새로운 Socket을 통해 연결이 수립되면 데이터를 송수신한다(send/rcv)
6. 모든 처리가 완료되면 Socket을 닫는다(close)

3. Socket Error

- 10050: 네트워크를 사용할 수 없기 때문에 소켓 작업을 진행할 수 없는 경우
- 10055: 버퍼가 부족하여 소켓에서 해당 작업을 진행하지 못한 경우
 - ⇒ 임시 버퍼에 메시지를 저장 후, 일정 시간 뒤 다시 메시지를 보낸다.
- 10058: 해당 소켓이 종료되었으므로 데이터를 보내거나 받을 수 없는 경우
 - ⇒ client에서 sever로 보낼 경우 임시 버퍼에 메시지를 저장 후, 서버 소켓을 bind하여 다시 메시지를 보낸다.
- 10060: 연결된 구성원으로부터 응답이 없어 연결하지 못했거나, 호스트로부터 응답이 없어 연결이 끊어진 경우
 - ⇒ Server socket을 새로운 port와 bind하여 통신 시도를 한다.

