

# LTS 서버

새로운 기술에 대한 호기심을 바탕으로 LTS 서버 구축에 도전하였고, 웹소켓 기술 및 통신 과정에 대해 자체적으로 학습하여 성공적으로 서버를 개발한 경험이 있습니다.

LH와의 협업으로, 업체별 서버의 통신 방법이 통일되지 않아 충전기와의 통신을 위한 중계 서버 개발이 필요로 하는 상황이었습니다. 비록 담당하는 업무가 아니었음에도 통신 관련 기술에 평소 관심이 많아 팀장님께 요청하여 해당 업무를 받아들일 수 있었습니다. 새로운 업무에 대한 도전이었기 때문에 프로세스 및 통신 관련 학습을 우선적으로 진행하였고 최종 개발을 완료하였습니다.

## [꼼꼼한 프로세스 정리를 통한 효율적 업무 진행]

먼저, 서버 프로그램이 진행되는 과정에 집중하여 학습을 진행하였습니다. LH 측에서 보내준 프로토콜 문서 분석을 통하여 전반적인 로직을 파악하였습니다. 서버가 할 일과 통신 방식에 초점을 맞추어 뼈대를 설계하였고 해당 과정에서 필요한 통신 방법에 대해 정리하였습니다. 이 과정에서 충전기와의 통신 프로그램과 LH 서버와의 통신 프로그램, 두 가지가 필요하다는 것을 확인하였고 express를 활용해 충전기와의 통신을 마무리할 수 있었습니다. 하지만 LH 측과의 통신에 필요한 웹소켓 방식은 생소하여 해당 방식에 대한 학습을 진행하였습니다.

## [실험을 통한 명확한 이해]

서버와의 통신을 위해서는 크게 웹소켓 통신 방식과 핸드셰이크 과정을 학습할 필요가 있었습니다. 따라서 각각의 정의를 확인함으로써 개념을 이해하고 실험 과정을 통해 해당 기술의 구현을 확인한 후 실제 프로세스에 적용하는 과정을 통해 실수없이 꼼꼼히 적용하는 과정을 거쳤습니다.

먼저 웹소켓은 tcp 통신의 단방향성이라는 단점을 보완하여 양방향 통신을 목적으로 만들어진 통신 방식입니다. 개발 목표는 client에서 서버로의 통신을 목적으로 만들어야 했으나 정확한 이해를 위해 서버-client 웹소켓 통신을 생성, 이를 수정해가며 통신 방식과 진행 과정을 하나하나 확인하는 과정을 거쳤습니다. 해당 과정에서 WS / Socket.io 두 가지 패키지를 활용해 진행하였고, 적용 난이도 및 확장성의 이유로 Socket.io를 이용해 개발완료하였습니다.

이후 인증서 확인 단계를 위한 핸드셰이크 과정을 학습하였습니다. 우선 개념정리부터 시작하였고 통신 과정에서 정상적인 통신을 위해 서로 간의 인증서 등을 교환하여 연결을 성사시키는 과정이라는 것을 알 수 있었습니다. 또한 핸드셰이크 과정에서 Socket.io를 통해 보낼 때 적용되는 옵션은 공식 사이트를 참조하여 Wireshark를 활용, 실제 소켓 통신 과정을 분석함으로써 핸드셰이크 과정을 정확히 이해하였습니다.

꼼꼼히 진행된 학습과 실험을 바탕으로 통신 프로그램을 작성하여 성공적으로 프로젝트를 마무리할 수 있었습니다. 기존 담당자가 아니었음에도 2주 간의 개발 기간 중 6일만에 프로젝트를 완성하여 전체 업무 진행에 큰 도움이 되었다고 자부합니다.