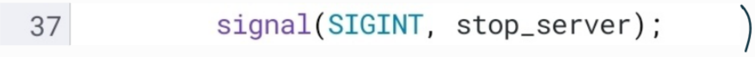
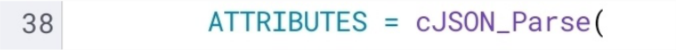
main 함수



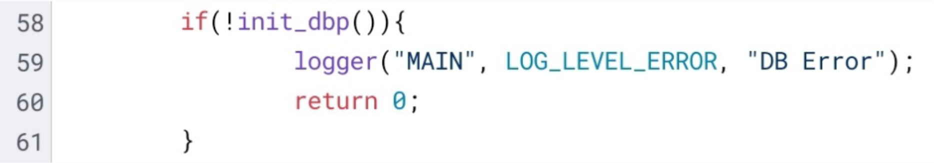
```
        signal(SIGINT, stop_server);
```

) Ctrl+C 누르면 stop_server 호출하여 종료



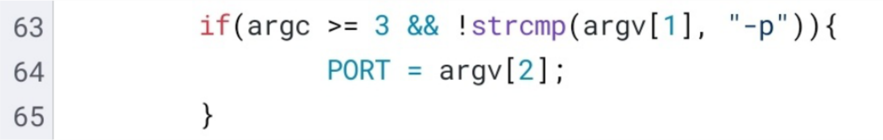
```
        ATTRIBUTES = cJSON_Parse(
```

cJson 라이브러리를 통해 JSON 문자열 파싱



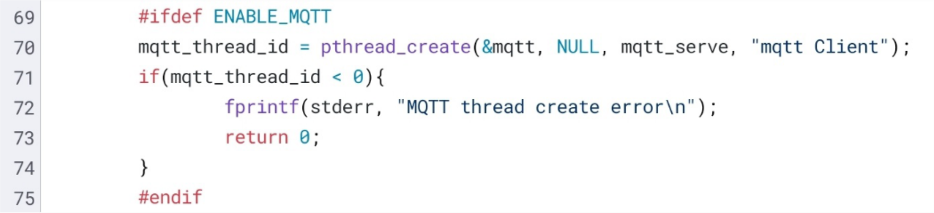
```
        if(!init_dbp()){
                logger("MAIN", LOG_LEVEL_ERROR, "DB Error");
                return 0;
        }
```

DB Error 가 있는지 확인



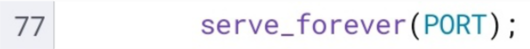
```
        if(argc >= 3 && !strcmp(argv[1], "-p")){
                PORT = argv[2];
        }
```

입력받은 인자가 3개 이상이고 2번째 인자가 "-p"일 경우 port 번호는 세번째 인자로 한다.



```
        #ifdef ENABLE_MQTT
        mqtt_thread_id = pthread_create(&mqtt, NULL, mqtt_serve, "mqtt Client");
        if(mqtt_thread_id < 0){
                fprintf(stderr, "MQTT thread create error\n");
                return 0;
        }
        #endif
```

만약 ENABLE_MQTT 가 정의되어 있으면 mqtt_serve 함수를 새로운 thread에서 실행한다.

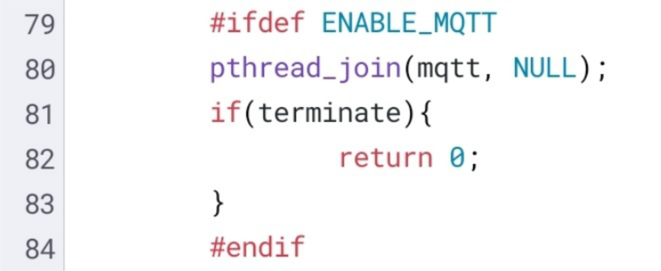


```
        serve_forever(PORT);
```

↳ serve_forever(PORT) 함수 호출

serve_forever 함수는 httpd.h 에 내장된 함수

↳ serve_forever 함수 안에 있는 start_server 함수가 서버를 시작함.

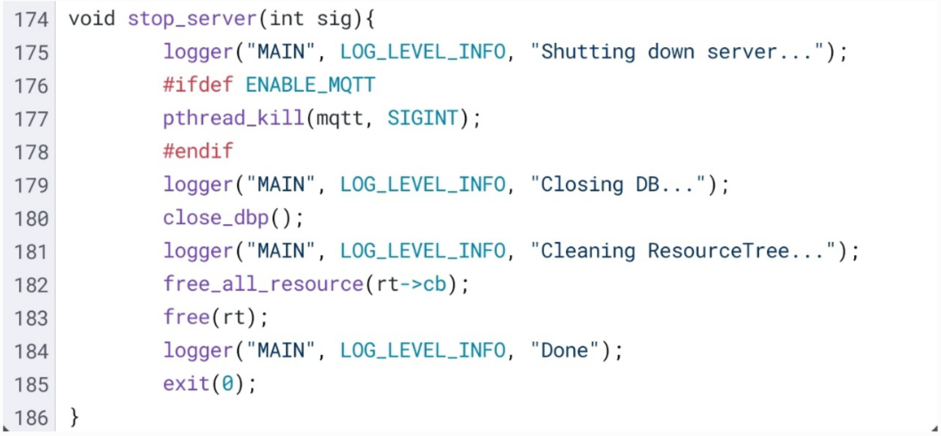


```
        #ifdef ENABLE_MQTT
        pthread_join(mqtt, NULL);
        if(terminate){
                return 0;
        }
        #endif
```

↳ pthread_join 함수는 thread가 종료하기를 기다리는 함수이다.
따라서 이 코드는 MQTT가 활성화되어 있을 때 thread가 종료되기를 기다리고, 종료되었을 시 return 하는 코드이다.

}

route 함수, handle_onemam_request // httpd.h 내장함수



```c
void stop_server(int sig){
        logger("MAIN", LOG_LEVEL_INFO, "Shutting down server...");
        #ifdef ENABLE_MQTT
        pthread_kill(mqtt, SIGINT);
        #endif
        logger("MAIN", LOG_LEVEL_INFO, "Closing DB...");
        close_dbp();
        logger("MAIN", LOG_LEVEL_INFO, "Cleaning ResourceTree...");
        free_all_resource(rt->cb);
        free(rt);
        logger("MAIN", LOG_LEVEL_INFO, "Done");
        exit(0);
}
```

↳ 서버 종료 함수