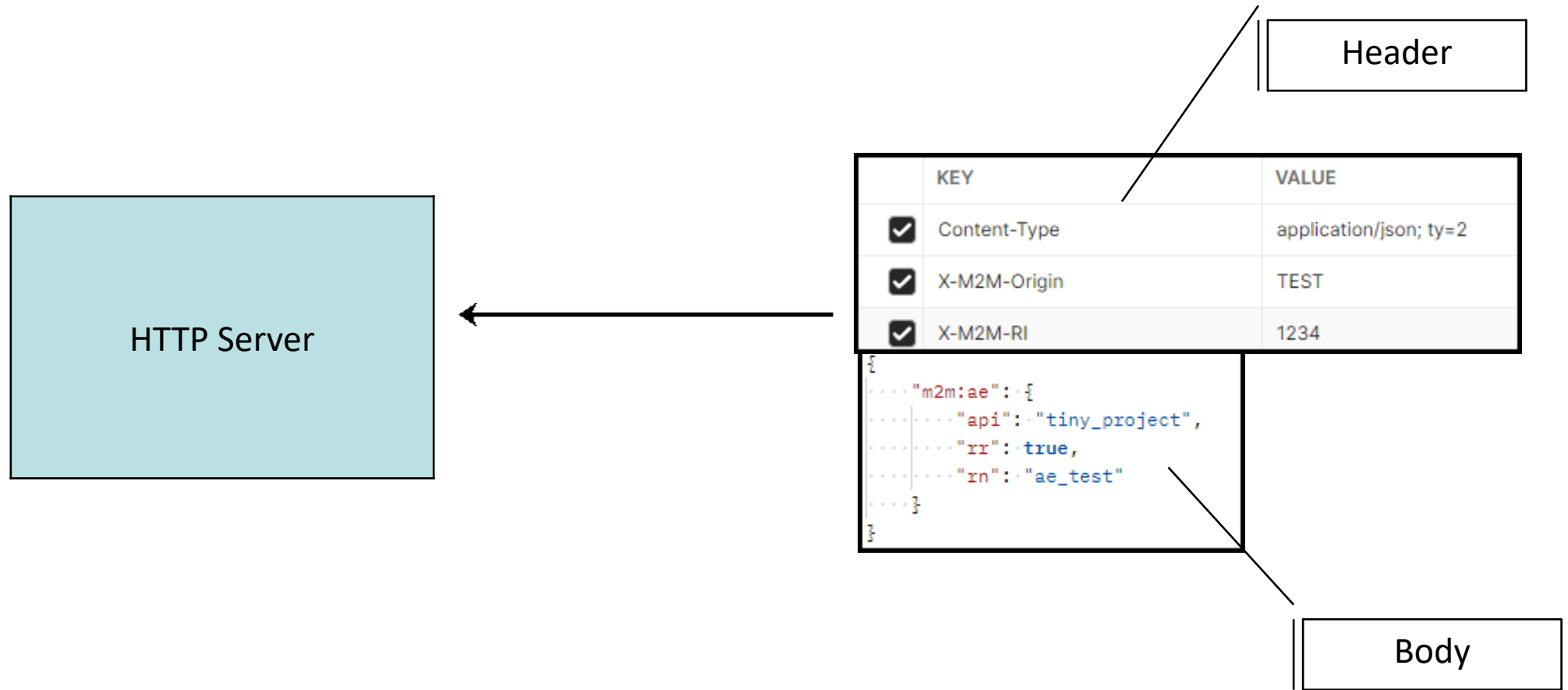


postman



1. Postman에서 다음과 같은 request를 서버로 보냅니다.

main.c

```
POST("/") {
    HTTP_201;
    if (request_header("X-M2M-Origin") && request_header("X-M2M-RI")) {
        if (payload_size > 0){
            char json_payload[payload_size];
            int index = 0;
            for(int i=0; i<payload_size; i++) {
                if(payload[i] != 0 && payload[i] != 32 && payload[i] != 10) {
                    json_payload[index++] = payload[i];
                }
            }
            json_payload[index] = '\0';
            printf("%s\n", json_payload);
        }
    }
    else
        printf("request is not OneM2M Standard.");
}
```

request_header는 인자로 전달된 필드가 존재하면 그 값을 반환해주는 함수입니다. 디폴트 값은 NULL입니다.

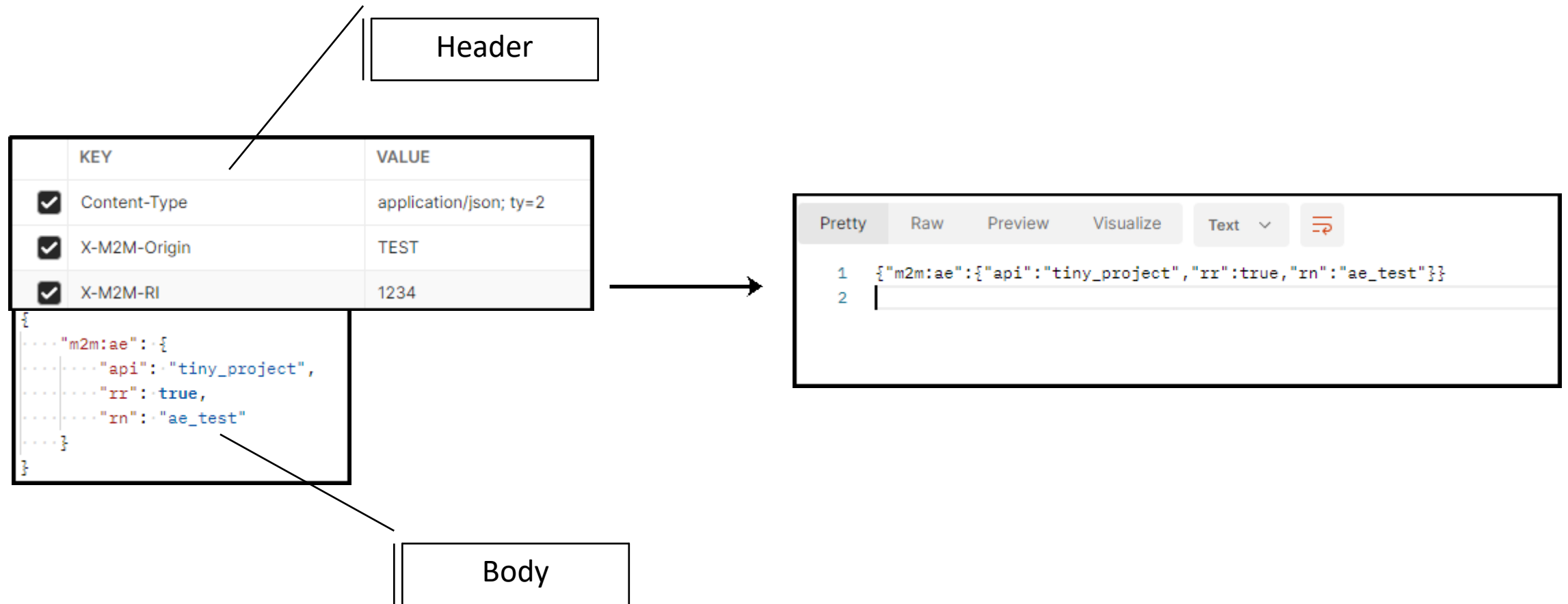
payload에 request body 값이 저장되어 있습니다. 다만 값의 일관성을 위해 NULL, SPACE, 개행을 제외한 값을 따로 다시 저장했습니다.

소스 코드 상 stdout은 clientfd와 바인딩되어 있어서 해당 출력은 client response에서 확인이 가능합니다.

OneM2M 규격이 아닌 경우입니다.

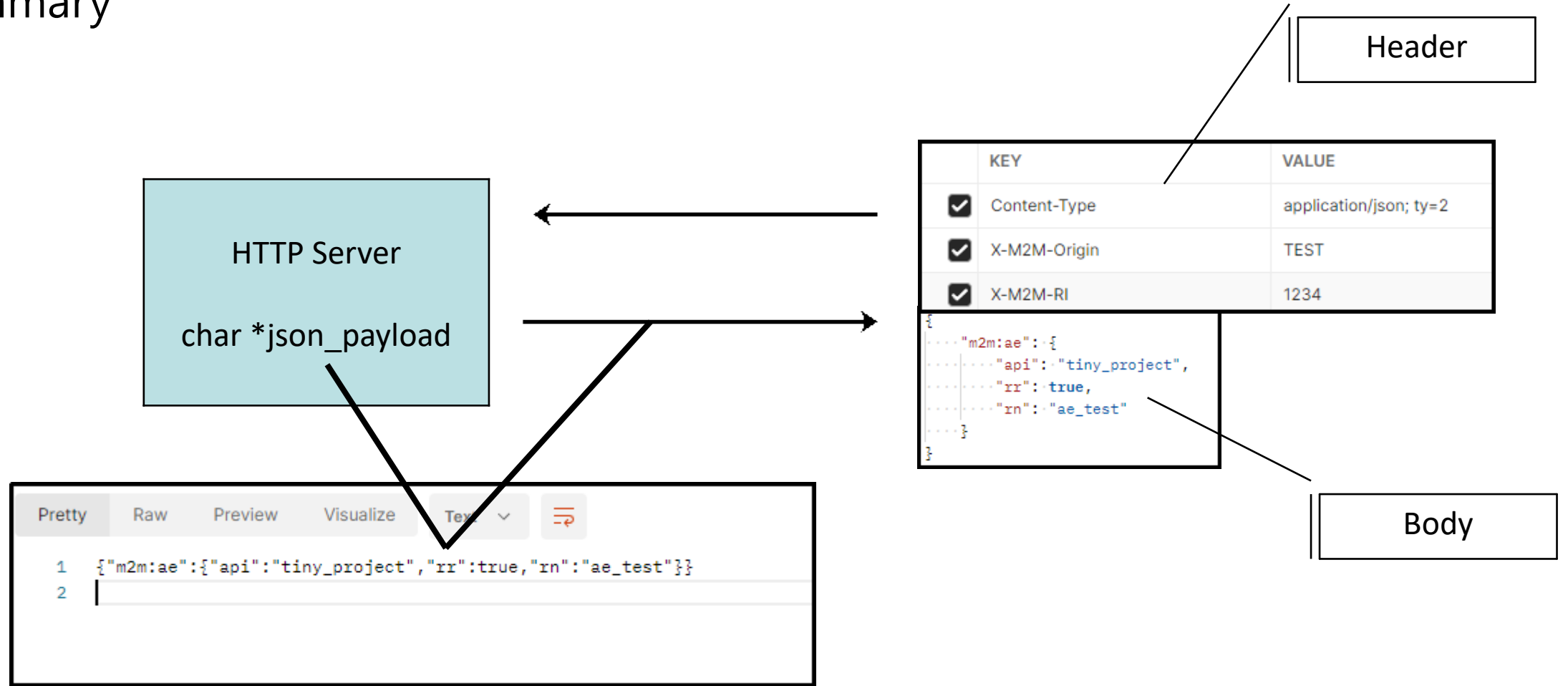
2. 서버 main.c에 해당하는 소스 코드입니다.

response



3. 해당 패킷에 대해서는 다음과 같은 응답을 기대할 수 있습니다.

summary



4. 다음과 같은 그림으로 정리가 가능합니다.

char* json_payload

```
eom@eom-virtual-machine:~/Desktop/Sejong-IoT-Project/OpenSource/pico$ ./server  
Server started http://127.0.0.1:3000  
{"m2m:ae":{"api":"tiny_project","rr":true,"rn":"ae_test"}}
```

5. char* json_payload값을 서버 로그에 찍어본 결과입니다.