[오픈소스 팀 프로젝트 일지\_4회차]

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 일시 | 2019년 12월 3일 16:00 ~ 24:00 | | | |
| 팀원 | 이다은 | 2019036380 | 임단비 | 2019056462 |
| 토의내용 | 1. **웹 서버 구현**   XAMPP를 통해 서버를 여는 것을 우선적으로 한다.  <http://alldpublic.kr/THFAPIs/apiControl.php?query=gdd&device=1>   * 데이터 불러오기 주소   <http://alldpublic.kr/THFAPIs/apiControl.php?query=udd&device=1&header=udd&data=ON>   * 데이터 추가하기 (바꿔야 하는 건 data 뒷부분 ON 또는 OFF)   alldpublic.kr(서버 주소)  THEAPIs(API가 담겨있는 곳, 디렉토리)  apiControl.php(THEAPIs안에 담겨있는 파일)  -업로드 담당하는 파일 불러오는 역할  -조회하는 파일 불러오는 역할  -두 기능을 합친 이유 –> 보안적으로 안전하게 보호하기 위해서  query=gdd&device(query에 따라서 기능을 분배해주는 역할)   * GET방식   : query의 뒷부분이 공개됨 (query=udd&device=1&header=udd&data=ON)   * POST방식   : query의 뒷부분이 공개되지 않음 (query=udd&device=1&header=udd&data=ON) 부분이 보이지 않음  Query(key)=udd(value)   * apiControl.php 코드 참고      * Key “query”가 value(case) “udd”를 받으면   require\_once로 연결되어 module 폴더의 device.php파일을 불러온다.  (C언어의 include기능)   * "./module/device.php"   “.”은 상대주소로 apicontrol 폴더 안의 module 파일로 들어가고  그 안에 있는 device.php를 불러온다.   * echo 는 print하는 함수로 본다.   uploadDeviceData()함수를 출력한다.  uploadDeviceData함수는 device.php파일에서 볼 수 있다.  (아래 device.php 파일 확인)    UploadDeviceData()함수를 시행시켜 리턴값을 json\_encode($jsonObj)로 받는다.  불러온 쪽 기준으로 상대주소를 적어줘야 되기 때문에  위 파일에서 require\_once("./module/dbConfig.php"); 을 한 것을 확인할 수 있다.   * $stmt = $conn->prepare("INSERT INTO THF\_Data(device, header, data) VALUES(?,?,?)");   :연결통로를 의미하는 $conn 변수, prepare은 query를 날릴 준비를 한다.  :THF 테이블에 데이터를 device, header, data를 순서대로 넣는다. ???는 아래의 bind\_param에서 정의해준다.   * @$stmt->bind\_param("sss", $\_GET['device'], $\_GET['header'], $\_GET['data']);   : sss – stringX3 GET방식으로 보낸다.  $stmt->execute();  Execute -> 실행시킨다. 실행 후 값이 들어간다.  <device=1&header=udd&data=ON>  Device 는 1, header는 udd, data는 ON으로 서버주소에서 나온다.  $stmt = $conn->prepare("SELECT COUNT(\*) FROM THF\_Data WHERE device=? and header=? and data=?");  @$stmt->bind\_param("sss",$\_GET['device'], $\_GET['header'], $\_GET['data']);  $stmt->execute();  $res = $stmt->get\_result();   * 데이터 무결성 검증과정 * 데이터는 변조되어 있는게 있으면 안됨 * Count는 삽입된 데이터의 똑 같은 데이터의 개수를 찾아준다. * $res = $stmt->get\_result();는 실행시킨 것의 결과를 가져오는 함수이다.   $row = mysqli\_fetch\_assoc($res);   * res에서 한 줄 씩 불러와야 하는데 데이터가 어차피 한줄이기에 바로 res라는 객체를 하나로 가져오면서 배열처럼 쓸 수 있다. * 출력구조는 jsonObj 자료구조를 따른다.   jsonObj 자료구조로 출력된 결과값  :{"ResCode":200,"ResMsg":"GET SUCCESS","Device":"1","Header":"udd","Data":"ON"}  if($row['COUNT(\*)'] == 0){  $jsonObj += ['ResCode'=>400, 'ResMsg'=>'UPLOAD FAIL'];  }  else if($row['COUNT(\*)'] > 0){  $jsonObj += ['ResCode'=>200, 'ResMsg'=>'UPLOAD SUCCESS'];  }   * $jsonObj += ['ResCode'=>400, 'ResMsg'=>'UPLOAD FAIL'];   :jsonObj 배열에 자료를 넣는 방법   * 만약 개수가 0개 이면 UPOAD FAIL -> 사람들에게 FAIL정보를 알려줌 * 개수가 0개 보다 크면 UPLOAD SUCCESS 하여 json\_encode함수를 불러와서 jsonObj로 결과를 보여준다.     결과가 출력 완료되면 다시 apiControl.php 로 돌아온다.  출력이 완료 되었으므로 break를 만나 udd 조건문을 빠져나와 gdd 조건문으로 들어간다.  gdd 조건문을 만나면 device.php에 있는 getDeviceData() 함수로 들어간다.  (아래 device.php 파일 확인)    위의 uploadDeviceData()함수와 비슷한 맥락이다.  "SELECT \* FROM THF\_Data WHERE device=? ORDER BY no desc LIMIT 1"   * 몇 번 디바이스(내가 설정하는 것)의 결과값인지를 출력해준다. * number을 기준으로ORDER BY 정렬을 해준다(내림차순으로)   @$stmt->bind\_param("s",$\_GET['device']);  $stmt->execute();  $res = $stmt->get\_result();   * 불러온 뒤 execute로 실행시키고, get\_result로 결과를 출력한다.   $row = mysqli\_fetch\_assoc($res);   * mySQL 실행 결과값을 배열로 만든다.   if($row2['COUNT(\*)'] == 0){  $jsonObj += ['ResCode'=>400, 'ResMsg'=>'GET FAIL'];  }  else if($row2['COUNT(\*)'] > 0){  $jsonObj += ['ResCode'=>200, 'ResMsg'=>'GET SUCCESS', 'Device'=>$row['device'], 'Header'=>$row['header'], 'Data'=>$row['data']];  }   * count 가 0이면 data가 없는 것이기 때문에 GET FAIL을 출력한다. * count가 0보다 크면 GET SUCCESS하여 device 번호와 data를 불러와서   json\_encode로 알려준다. | | | |
| 결론 | 다음시간에   * 진동센서가 진동을 감지하면 웹화면에 그 결과를 출력하는 코드 완성하기 * 웹 화면 HTML로 구현 완료하여 화면에 진동센서의 결과값을 출력하는 과정 완성하기 | | | |