

PHP REST API (1)

이미 공지한 대로

- 이 수업은 이러닝에 업로드한 SQL 강의를 들었어야 진행이 된다
- 듣지 않은 학생은, 이 수업 녹화본 역시 자료실에 업로드할 예정
이므로, SQL 강의를 들은 후에 본 강의를 학습할 것

이러닝 자료실에 가서,

- 오늘 할 수업코드를 받아 bitnami htdocs 안에 담자

test 라는 이름의 db 만들기

test 라는 이름의 db를 만들 것이므로,
일단 있는지부터 확인해보자.

```
MariaDB [(none)]> SHOW DATABASES;  
+-----+  
| Database |  
+-----+  
| information_schema |  
| lunchbox |  
| mysql |  
| performance_schema |  
| test |  
+-----+  
5 rows in set (0.001 sec)  
  
MariaDB [(none)]> |
```

있으면, 이 db를 날려버리도록 하자.
날리기 싫으면 다른 이름으로 지어도 된다.

```
MariaDB [(none)]> DROP DATABASE test;  
Query OK, 0 rows affected (0.001 sec)  
  
MariaDB [(none)]> |
```

그리고, 다음 명령어로 test 라는 이름의 db를 만든다.

```
MariaDB [(none)]> CREATE DATABASE test;  
Query OK, 1 row affected (0.001 sec)  
  
MariaDB [(none)]> |
```

Table Plus 로 연결

MariaDB Connection

Name: 웹프실습 php 테스트디베

Status color: ☐ ☐ ☒ ☐ ☐ ☐ Tag: local

Host: localhost Port: 3306

User: root

Password:

Database: test Bootstrap commands...

SSL mode: PREFERRED Clear keys

Key... Cert... CA Cert...

☐ Over SSH

Server: Port:

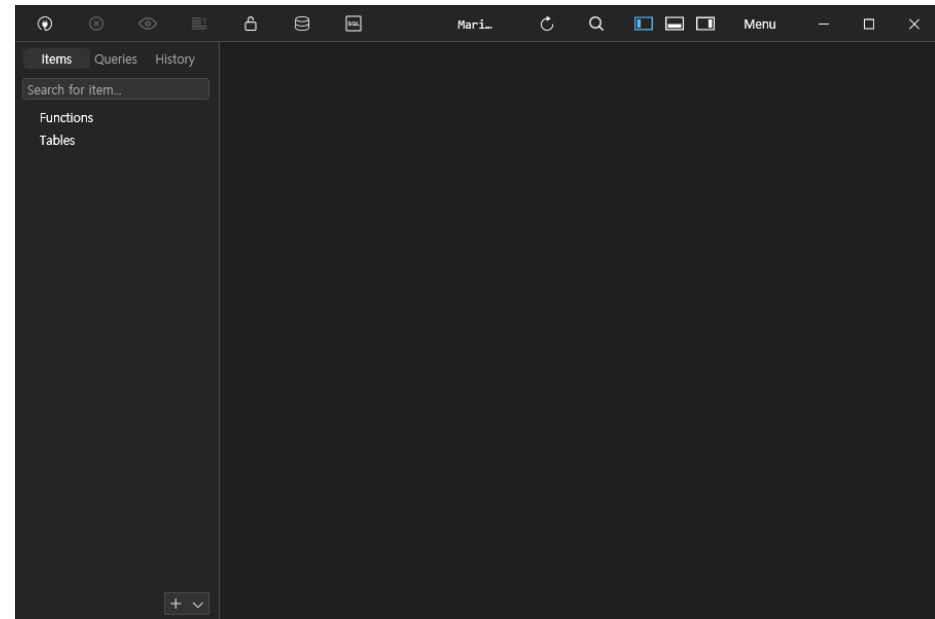
User:

Password:

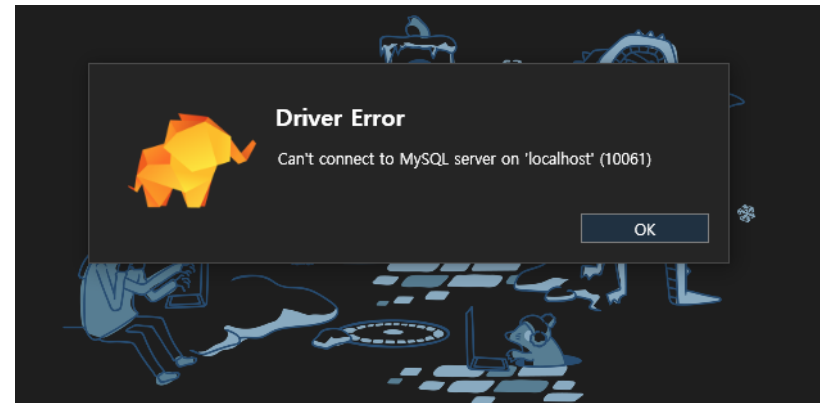
☐ Use SSH Key SSH private key... Passphrase:

Save Test Connect

성공하면 다음과 같이 로딩 후 빈 화면이 뜬

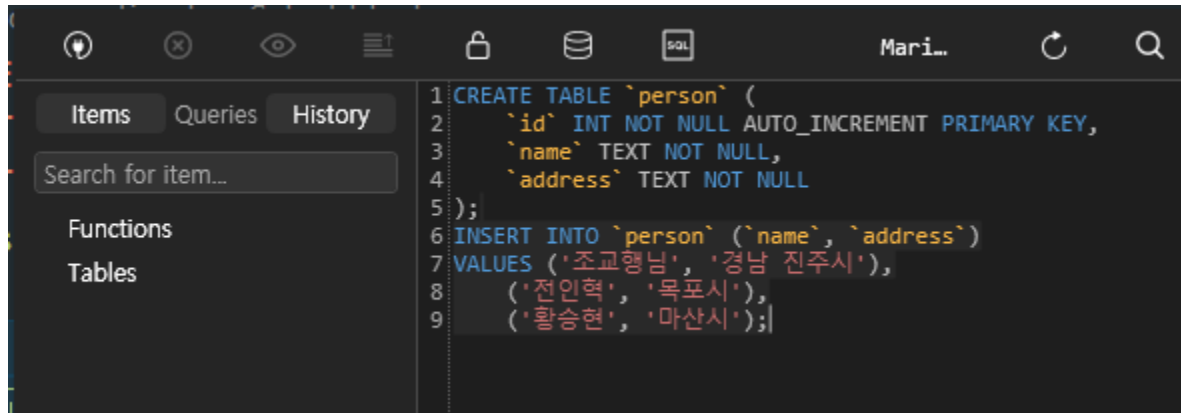


연결 실패 시 에러메세지



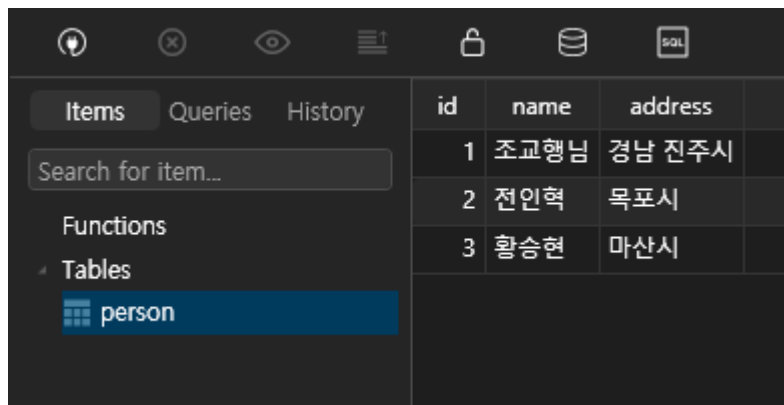
DB Table 만들기

db.sql 파일 안에 있는 코드를 TablePlus 에서 "한 문단씩" 실행(ctrl + enter)



The screenshot shows the TablePlus SQL editor interface. On the left, there's a sidebar with 'Items', 'Queries', and 'History' tabs. Below them is a search bar and a 'Tables' section. The main area displays SQL code for creating a table and inserting data.

```
1 CREATE TABLE `person` (  
2   `id` INT NOT NULL AUTO_INCREMENT PRIMARY KEY,  
3   `name` TEXT NOT NULL,  
4   `address` TEXT NOT NULL  
5 );  
6 INSERT INTO `person` (`name`, `address`)  
7 VALUES ('조교행님', '경남 진주시'),  
8         ('전인혁', '목포시'),  
9         ('황승현', '마산시');
```



The screenshot shows the TablePlus interface with the 'person' table selected in the 'Tables' sidebar. The table data is displayed in a grid view.

| id | name | address |
|----|------|---------|
| 1 | 조교행님 | 경남 진주시 |
| 2 | 전인혁 | 목포시 |
| 3 | 황승현 | 마산시 |

여기까지 했으면, DB 준비는 다 된 것

selectUser.js

- 이 파일엔 두 개의 함수가 있다.
 - getAllUser() 모든 유저를 가져옴
 - getAddressByID() ID와 일치하는 유저의 주소를 가져옴
- 둘 다 onclick 이벤트핸들러에 의해 실행될 것

selectUser.html

```
<body>
  <h1>유저 조회</h1>
  <h2>모든 유저의 정보를 가져옵니다.</h2>
  <button onclick="getAllUser()">모든 유저 가져오기</button>
  <h1>이름과 일치하는 유저의 주소를 가져옵니다.</h1>
  <input class="nameInput" type="text" placeholder="이름">
  <button onclick="getAddressByID()">이름과 일치하는 유저 주소 가져오기</button>
</body>
```

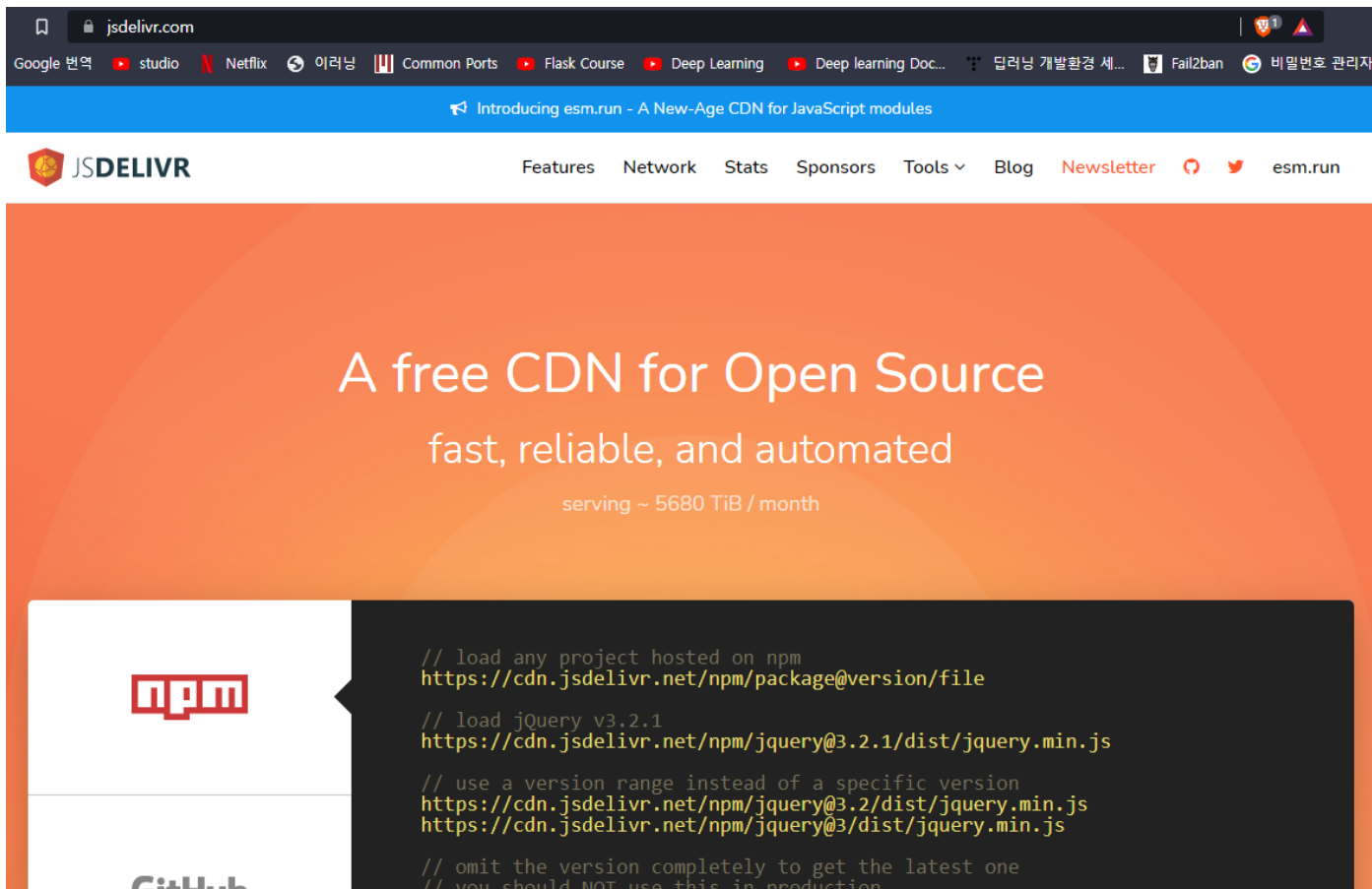
selectUser.js

```
const getAllUser = () => {
  console.log("getAllUser 함수 실행");
};

const getAddressByID = () => {
  console.log("getAddressByID 함수 실행");
};
```


서버의 API (PHP 파일) 을 실행하려면


- JavaScript 에서 fetch() 함수를 제공
- 그러나, 현재 업계에서 표준은 axios 라는 패키지
 - 패키지:
 - 개발자 선배들이 "미리 만들어놓은" 코드
 - 우리가 모든 걸 다 개발할 필요는 없다
 - 라이브러리, API 와 비슷한 뜻. 세 가지 뜻 모두, 개발자들이 미리 만들어둔 코드 라는 뜻을 가지고 있다.






jsdelivr 라는 사이트에서
각종 패키지를 CDN 으로 제공

CDN:
패키지가 저장된 링크
직접 설치하기 싫을 때 유용함

Q axios

Real-time search by  algolia

axios

 axios  0.23.0  MIT


Promise based HTTP client for the browser and node.js

[xhr](#) [http](#) [ajax](#) [promise](#) [node](#)



axios




 axios  0.23.0  MIT  0 vulnerabilities

Promise based HTTP client for the browser and node.js

[xhr](#) [http](#) [ajax](#) [promise](#) [node](#)

axios CDN Files

axios


0.23.0 

Default file: /dist/axios.min.js

Copy to Clipboard

 /npm/axios@0.23.0/dist/axios.min.js



 /npm/axios@0.23.0/dist

 /npm/axios@0.23.0/lib

 /npm/axios@0.23.0/CHANGELOG.md

 /npm/axios@0.23.0/index.d.ts

Copy URL

Copy SRI

Copy HTML

Copy HTML + SRI



selectUser.html

```
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <!-- axios -->
  <script src="https://cdn.jsdelivr.net/npm/axios@0.23.0/dist/axios.min.js"></script>

  <script src="./selectUser.js"></script>
  <title>유저 조회</title>
</head>
```

<head> 태그 안에 복사한 axios html cdn 을 붙여넣자.

JS selectUser.js ●

JS selectUser.js > [x] getAddressByID

```
1  const getAllUser = () => {  
2    try {  
3  
4    } catch (error) {  
5      console.log(error);  
6    }  
7  };
```

try catch 는 통신 에러 발생 시 실행되는 에러처리구문이다.
서버가 트래픽이 과도하게 발생하거나, 셧다운되거나,
원하는 경로에 파일이 없거나 하는 문제가 발생하면
catch 에서 에러를 받아 적절한 일을 한다

ex) 경고창: 사이트에 문제가 있습니다. 잠시 후 다시 시도해주세요

우리가 실행시키고자 하는 구문은 try 에 작성한다.

```
JS selectUser.js X
JS selectUser.js > ...
1  const getAllUser = () => {
2    try {
3      const response = axios.get("http://203.255.3.246:9000/getAllUser.php");
4      if (response.data) {
5        console.log(response.data);
6      }
7    } catch (error) {
8      console.log(error);
9    }
10 };

```

get 과 post 방식은 조금 있다가 자세하게 설명하고, 일단 쪽 입력한다.

<http://203.255.3.246:9000/getAllUser.php>

당장 필요한 지식만 간단히 설명하자면,

axios.get() 을 사용해 getAllUser.php 파일에 접근한다.

이 php 파일은 어떤 DB의 모든 유저데이터를 가져오는 일을 하며, response 객체를 리턴한다.

response 객체 안에 data 가 바로 getAllUser.php 가 한 일의 결과다.

그걸 console.log() 로 찍어내보는 것이다.

그러나,

- 버튼을 눌러보면 아무것도 콘솔에 안 찍힐 것이다.
 - 함수 이름을 잘못 썼나? 그럴 리 없다. 콘솔로그로 테스트했다.
 - 심지어 에러 로그도 뜨지 않는다.
- 당연히 아무것도 안 찍힌다. 여러분은 비동기의 개념을 모르기 때문이다.

비동기(Asynchronous)

이게 실행되기도 전에 바로 아래에 있는 if 가 실행되어버림
왜? JavaScript 는 건방지게도,
늦어질 것 같으면 다음 행을 먼저 실행시켜버림

```
const getAllUser = () => {  
  try {  
    const response = axios.get("http://203.255.3.246:9000/getAllUser.php");  
    if (response.data) {  
      console.log(response.data);  
    }  
  } catch (error) {  
    console.log(error);  
  }  
};
```

```
if (response.data) {  
  console.log(response.data);  
}  
const response = axios.get("http://203.255.3.246:9000/getAllUser.php");
```

실제 실행된 순서. 애초에 response.data 안에 아무것도 없으니깐 실행이 안 되는 것
그럼 undefined 에러가 터야하지 않나? 그건 또 아닌게, response를 const 변수로 선언하긴 했음

왜 이런 끔찍한, 비동기방식을 쓸까?

- 만약 엄청나게 느린 서버가 있다고 가정해보자.
- 비동기가 없다면, 서버로부터 데이터가 올때까지 계속 기다려야함
 - 배달의민족에 떡볶이 주문버튼을 클릭했는데 20분을 기다리고싶은가?
- JavaScript 는 이런 사태를 방지하기위해, 서버와 통신하는 코드가 있을 땐 무조건 다음 코드를 먼저 실행시킨다.

해결책1 : 콜백함수

- 2015년 이전엔 유일한 해결책
- 파라미터로 들어와서 실행된 함수가 “끝나야만” 다음 로직 진행
- 이 경우, 콜백 지옥(callback hell) 에 빠질 수 있음
- JavaScript 개발자들이 가장 싫어하는 현상
- 여러분은 오른쪽과 같은 코딩을 하고 싶은가?

```
step1(function (value1) {  
  step2(function (value2) {  
    step3(function (value3) {  
      step4(function (value4) {  
        step5(function (value5) {  
          step6(function (value6) {  
            // Do something with value6  
          });  
        });  
      });  
    });  
  });  
});
```

해결책2: Promise 객체

- 고급 JavaScript 개발자라면 반드시 배워보길 바람
- pending, fulfilled, rejected 를 통해 비동기 제어
- 그러나 여전히 어렵다

해결책3: async await

- 매우 쉬움. 비동기를 동기적으로 제어하는 것
- async 로 비동기 로직이 포함되어있음을 알리고,
- await 로 끝날때까지 기다리라고 지시

async 비동기가 포함되어있음을 알림

```
const getAllUser = async () => {  
  try {  
    const response = await axios.get("http://203.255.3.246:9000/getAllUser.php");  
    if (response.data) {  
      console.log(response.data);  
    }  
  } catch (error) {  
    console.log(error);  
  }  
};
```

await 이게 끝나기전까진 다음줄로 넘어가지 마라!

결과:

```
▼ (3) [{...}, {...}, {...}] ⓘ  
  ▶ 0: {id: 1, name: '조교행님', address: '경남 진주시'}  
  ▶ 1: {id: 2, name: '전인혁', address: '목포시'}  
  ▶ 2: {id: 3, name: '황승현', address: '마산시'}  
    length: 3  
  ▶ [[Prototype]]: Array(0)
```

우리가 할 일

- 우리는 203.255.3.246 서버의, 9000번 포트의,
- 조교가 만들어둔 getAllUser.php 를 사용했다.
- 이제 직접 php api 를 만들 것이다.

JavaScript 의 한계

- DB CRUD 불가능
 - 사실은 틀린 말. Node.js 등장 이후 가능해짐
 - 그러나, DB CRUD 만 하기엔 Node.js 는 지나치게 어렵다
- 페이지를 벗어나면 데이터 보존이 안 됨
 - login.html 에서 main.html 로 넘어가는 순간, login.html 에서의 모든 데이터는 초기화되므로 main.html 로 데이터를 넘길 수 없음
 - 이것을 해결하려면 세션(session) 변수를 브라우저가 아닌 서버에 저장해야함

왜 PHP인가?

- 개인적으로, 2021년 웹 입문자라면 Python 을 쓰는 Django 를 배우길 추천
 - DB CRUD 를 시작하기 매우 좋음
- PHP는 1995년에 등장해 인터넷 부흥기를 이끈 언어
 - 무려 26년 역사
 - 이 정도면 웹 시장에선 영감님
- 그러나, Django 와 비교해봤을 때 DB CRUD가 어렵지는 않다
 - 특히, jsp Spring 과 비교해보면 엄청나게 쉬운 언어
 - 우린 PHP를 사용해 DB CRUD 를 하는 REST API 를 만들 것이다

REST 는 무엇인가

- 통신 규칙
- 누구도 부정할 수 없는 업계 표준.
 - 큰 회사든 작은 회사든, 오래된 회사든 스타트업이든,
 - 외국회사든, 한국회사든,
 - REST 라는 "규칙"을 지킨다.
- 2000년 로이 필딩(Roy Fielding) 의 박사학위 논문에 정의
- 기존 SOAP 라는 "규칙" 과 경쟁하다가 완전히 시장 장악

REST 는 무엇인가?

- 특정한 기술이 아니라 "규칙"
- HTTP를 사용해 데이터를 주고받는 "규칙"
- 우체국 송장처럼, 정해진 하나의 양식(form)
 - 보내는 분 주소는 받는분 성명에 쓰면 당연히 안된다.
 - 착불로 보내고 싶으면 체크박스에 체크해야한다.
- API 에서 데이터를 가져오고 싶으면 GET 방식
- API 에 데이터를 보내고 싶으면 POST 방식

대한민국 KOREA
신청 및 배달안내 1588-1300
마음까지 전하는 우체국 택배
우편번호가 2015년 8월1일부터 바뀝니다.
1110-1110 > 031817
www.ePOST.kr

| | | | | | | | | | | |
|------------------|----|--------|--|--|--|---|-------------------------------|---------------------------------|----|--|
| 보 내 는 분 | 성명 | | | | | 내 용 물 | | | | |
| | 전화 | | | | | *안심소포로 접수한 경우에는 표기하신 내용을 가액의 범위에서 (300만원 이내) 실 손해액을 배상받을 수 있습니다. (우편법 제38조) | | | | |
| | 주소 | □□□□□□ | | | | <input type="checkbox"/> 착불소포 | 원 | <input type="checkbox"/> 안심(보험) | 만원 | |
| | | | | | | <input type="checkbox"/> 파손주의 | <input type="checkbox"/> 취급주의 | <input type="checkbox"/> 대금교환 | 만원 | |

- 고객안내사항 -

- 부패, 변질, 파손(훼손)이 우려되는 내용물은 '특수포장' 할 경우에 한하여 접수 가능 합니다. 책임 원인이 발송인에게 있는 경우 손해배상에서 제외됩니다. (우편법 제39,40조)
- ※ 냉동 · 냉장물품은 보냉재(식용얼음 또는 아이스팩 등)를 넣어 포장하되, ① 육류는 별도 진공포장 ② 부패, 변질가능 우편물은 스티로폼 박스 사용

증지라벨 붙이는 곳 (우체국사용)

| | | | | | |
|-------------|----|--------|--|--|--|
| 받 는 분 | 성명 | | | | |
| | 전화 | | | | |
| | 주소 | □□□□□□ | | | |

개인정보 유출방지를 위하여 성명, 전화번호, 주소를 제거 바랍니다.

API 는 무엇인가

- 라이브러리, 패키지와 비슷한 뜻
- 쉽게 생각하면 특정한 일을 하는 코드다.
 - ctrl + c 를 누르면 복사가 된다는건, ctrl + c 를 "입력" 받아서 클립보드에 복사하는 일을 하는 코드가 "호출(call)" 된다는 것
- 우리가 만들어볼 현실적인 예
 - getAllUserData 라는 PHP 파일을 "호출"하면, DB에서 모든 유저의 데이터를 가져온다.

REST API 는 무엇인가

- REST 규칙을 따르는 API
- 그래서 우린, REST API 를 PHP 로 만들 것이다.

경대도시락의 예



우리가 해볼 것: php 파일 3개 만들기

createUser.php

유저 생성

getAllUser.php

모든 유저의 정보를 가져옵니다.

모든 유저 가져오기

```
▼ (3) [{...}, {...}, {...}] ⓘ  
  ▼ 0:  
    address: "서울시 분당구"  
    id: 29  
    name: "조교행님"  
    ▶ [[Prototype]]: Object  
  ▼ 1:  
    address: "전남 목포"  
    id: 30  
    name: "전인혁"  
    ▶ [[Prototype]]: Object  
  ▼ 2:  
    address: "부산광역시"  
    id: 31  
    name: "황혁주"  
    ▶ [[Prototype]]: Object  
length: 3  
▶ [[Prototype]]: Array(0)
```

getUserById.php

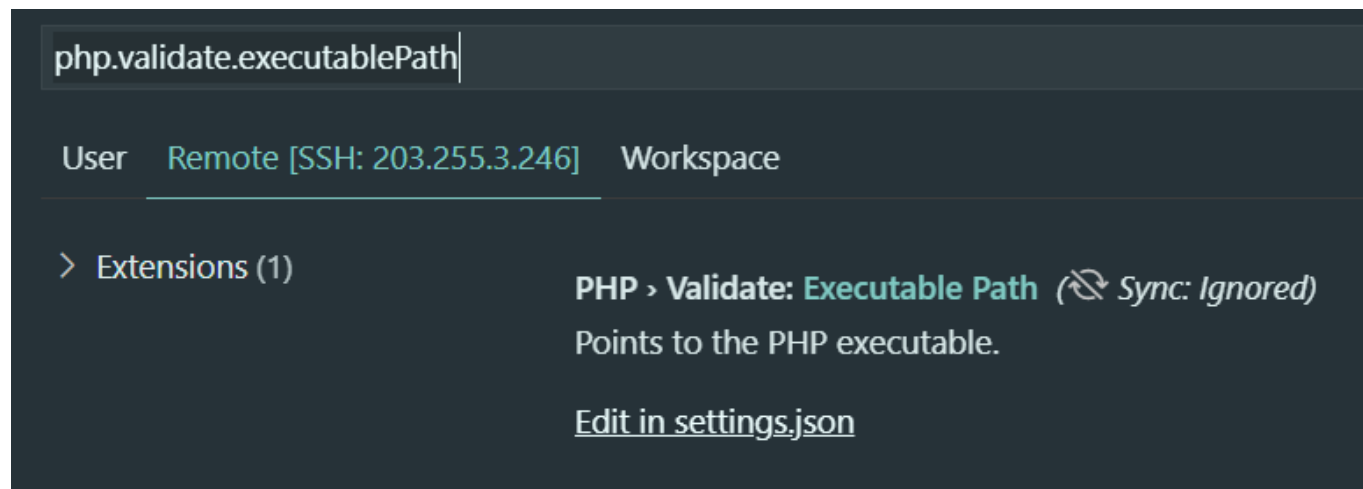
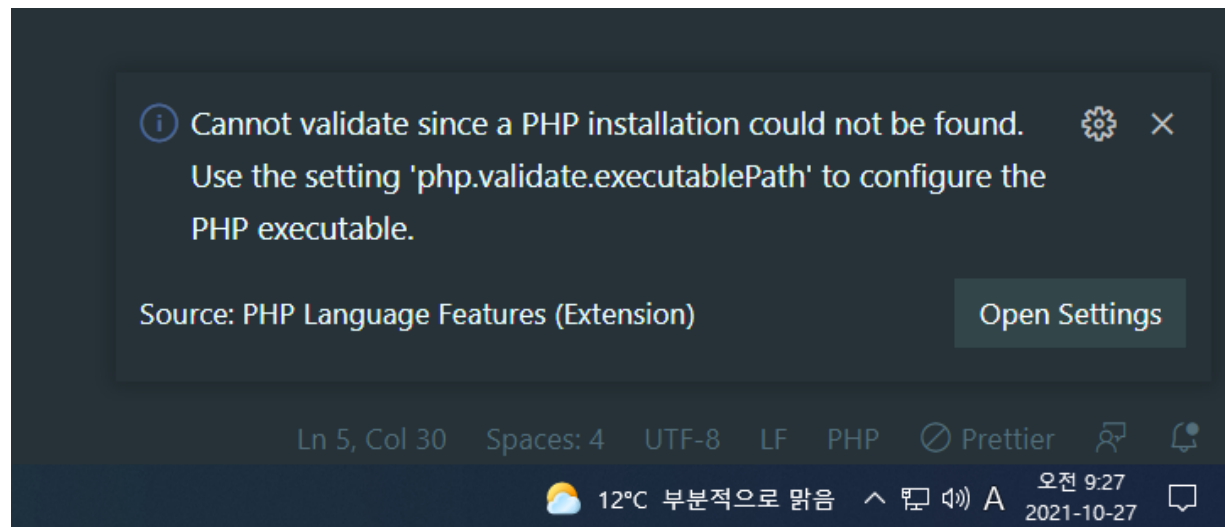
이름과 일치하는
유저의 주소를
가져옵니다.

조교행님

이름과 일치하는 유저 주소 가져오기

▶ {address: '서울시 분당구'}

php 환경 설정 (필수는 아님. Extension 잘 작동되게 하기 위함)



```
"php.validate.executablePath": "C:/Bitnami/wampstack-8.0.9-0/php/php.exe",
```

당연히, Mac 은 C:/ 가 아니다.