

学习笔记(6.29-laravel)

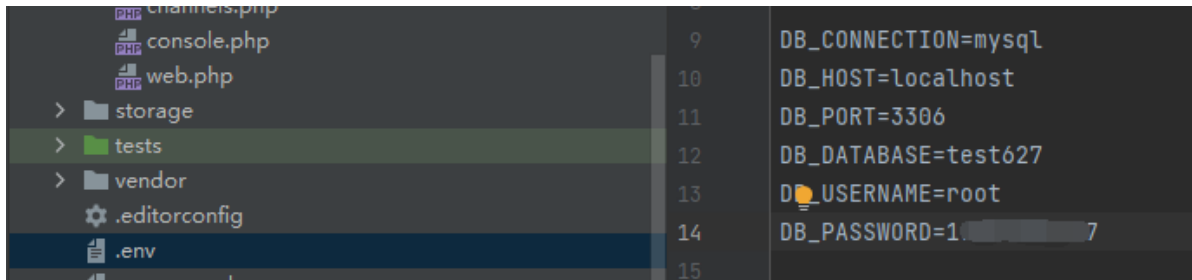
知识点

数据库

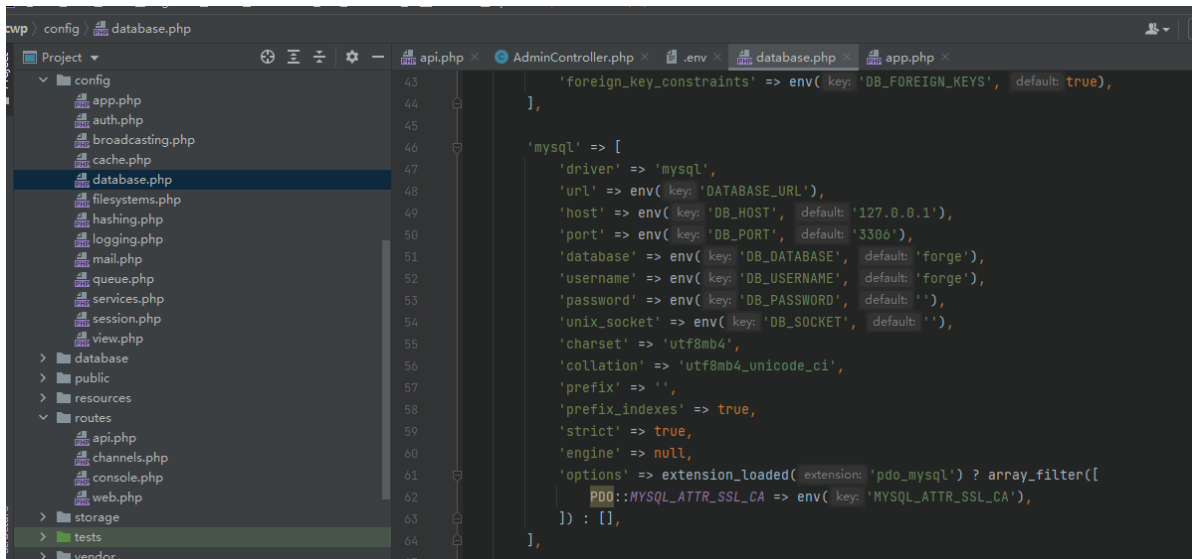
对数据的操作我们一般放在model层来完成，如果不使用model，则可以通过laravel框架提供的DB类操作数据库。而且对于一些极其复杂的sql，用model很难去完成，需要通过开发者自己手写sql语句，使用DB类执行原生sql。

其基本用法为：DB::table('表名')

数据库配置连接



推荐使用本地连接，使用ip地址是远程连接，可能不被mysql允许！



通过config目录下的database.php也可以修改mysql的配置信息，'strict'=>true严格模式建议关闭，utf8mb4包含了utf8，这里使用utf8mb4数据库创建用utf8不会出错。

如果是通过php artisan serve 来启动项目的话，修改配置文件后需将其重新启动，才能读取修改后的配置文件；

配置好后准备使用，要在控制器中引入DB门面，而DB门面在app.php中已经定义别名，可以直接use，不需要写太长。直接 **use DB;**

定义增删改查的路由

```
Providers
├── User.php
├── bootstrap
├── config
├── database
├── public
├── resources
├── routes
│   ├── api.php
│   ├── channels.php
│   ├── console.php
│   ├── web.php
│   ├── storage
│   ├── tests
│   └── vendor
└── ...

//DB下的
Route::prefix('test')->group(function () {
    //增加
    Route::get('add', function () {
        //删除
        Route::get('del', function () {
        //查询
        Route::get('update', function () {
        //修改
        Route::get('select', function () {
    });
});
```

注意：在api.php下定义的路由访问时url前面要加 api/

增加 (insert)

有两个函数可以实现表中增加数据，insert()和insertGetId()

insert(数组)可以同时添加一条或多条数据，返回值是布尔类型。

insertGetId(一维数组),只能添加一条数据，返回自增的id。

DB::table('表名')->insert(); 连贯操作/链式操作

127.0.0.1:8000/api/test/add

应用 百度 lookdiv Chapter 1 Test(C... c++查询库cppref... c++问题查询: Sta... 标准委员会, 好文... 官方文档

CS Window Help cwp - AdminController.php

api.php x AdminController.php x .env x database.php

```
29 }
30
31 //添加方法
32 public function add(){
33     //定义关联操作的表
34     $db = DB::table('member');
35     //使用insert增加记录
36     $result_add = $db->insert([
37         [
38             'name' => '假面佳欣',
39             'age' => '18',
40             'email' => 'jiamianjaxin@qq.com'
41         ],
42         [
43             'name' => '超人德志',
44             'age' => '18',
45             'email' => 'chaorendezhi@qq.com'
46         ]
47     ]);
48     dd($result_add);
49 }
50 }
```

member @test627 (zcr) - 表 - Navicat Premium

文件 查看 收藏夹 工具 窗口 帮助

连接 用户 表 视图 函数 事件 查询

zcr

book
information_schema
meal
mysql
performance_schema
registration
sys
test01
test02
test627

表

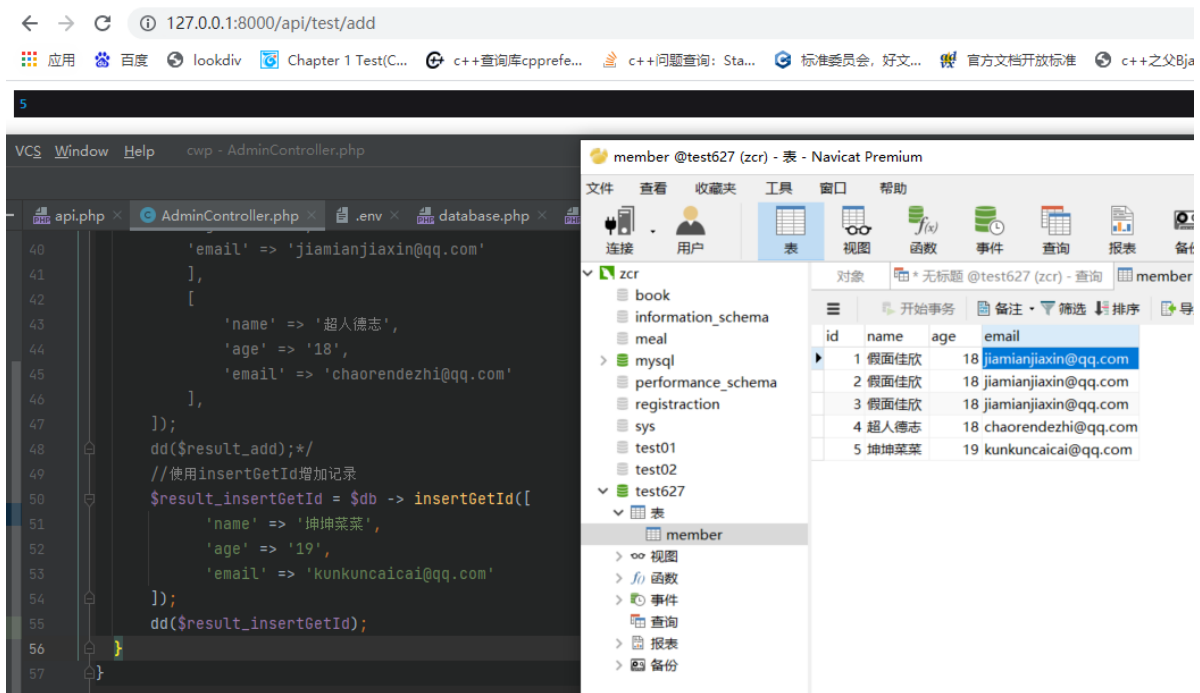
member

视图
函数
事件
查询
报表
备份

对象 * 无标题 @test627 (zcr) - 查

开始事务 备注 筛选

| id | name | age | email |
|----|------|-----|------------|
| 1 | 假面佳欣 | 18 | jiamianjia |
| 2 | 假面佳欣 | 18 | jiamianjia |
| 3 | 假面佳欣 | 18 | jiamianjia |
| 4 | 超人德志 | 18 | chaorend |



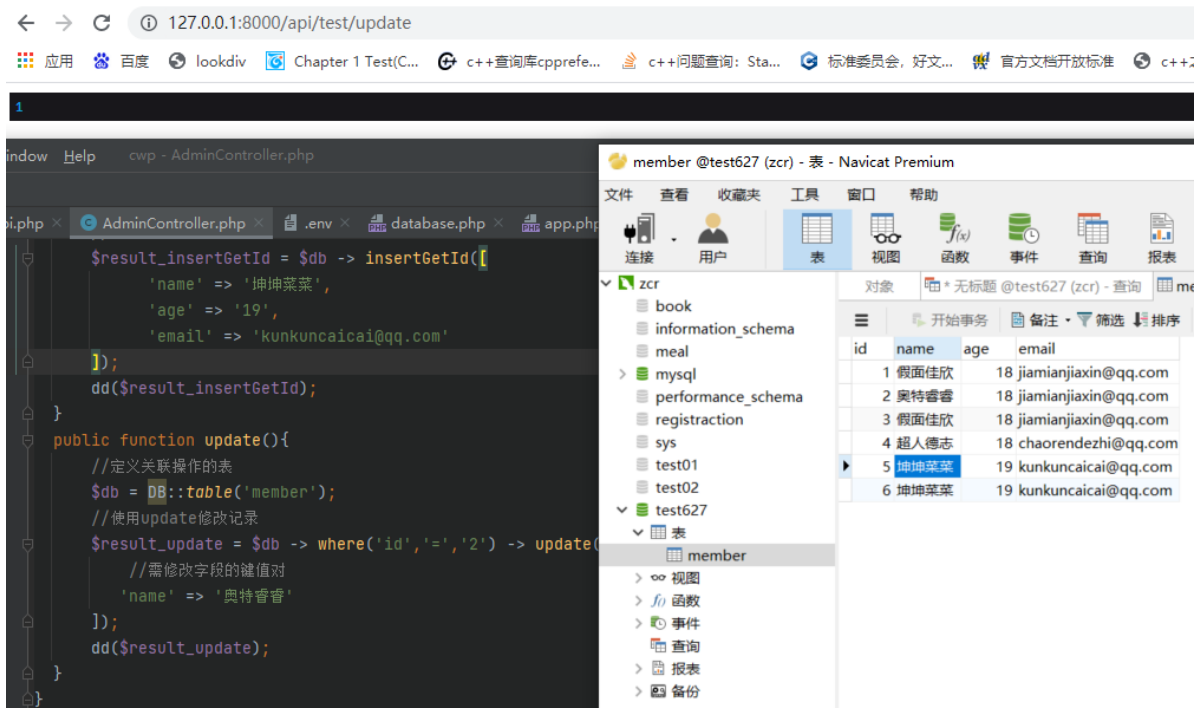
修改 (update)

修改数据可以用update(),increment()和decrement()函数来实现。

Update表示可以修改整个记录中的全部字段；

Increment 和 decrement 表示修改数字字段的数值（递增或者递减），典型应用：来记录登录此书，借书增加，积分增加；

格式：->where(字段, 运算符, 值)->update([]) 如果运算符为'='，则第二个参数可以不写。



注意：此时返回的结果为受到影响的行数。

```

$db = DB::table('member');
//把id=1的用户age字段，每次刷新
$result_increment = $db -> where('id','=', '1') -> increment('age'); //+1
$result_increment5 = $db -> where('id','=', '1') -> increment('age',5);

//+5
$result_decrement = $db -> where('id','=', '1') -> decrement('age'); //-1
$result_decrement5 = $db -> where('id','=', '1') -> decrement('age',5);

//-5

```

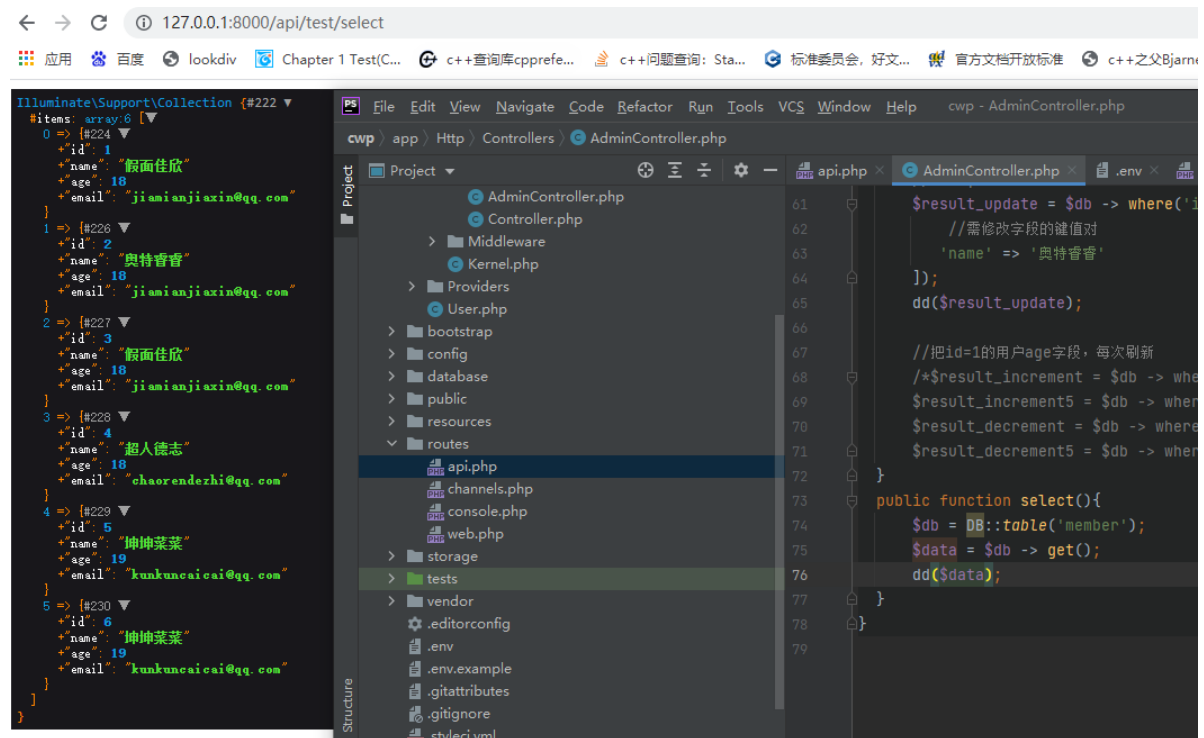
查询数据 (get)

- 取出基本数据

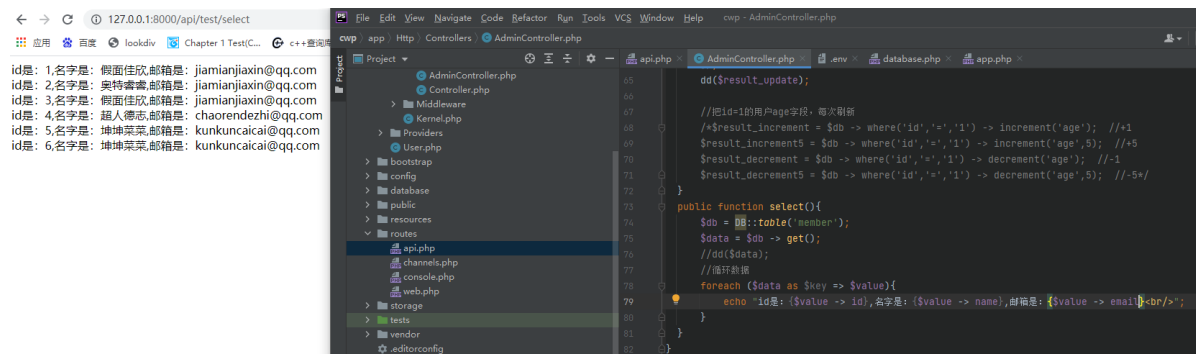
获取member 表中那个所有的数据

DB::table('member')->get(); //相当于 select * from member

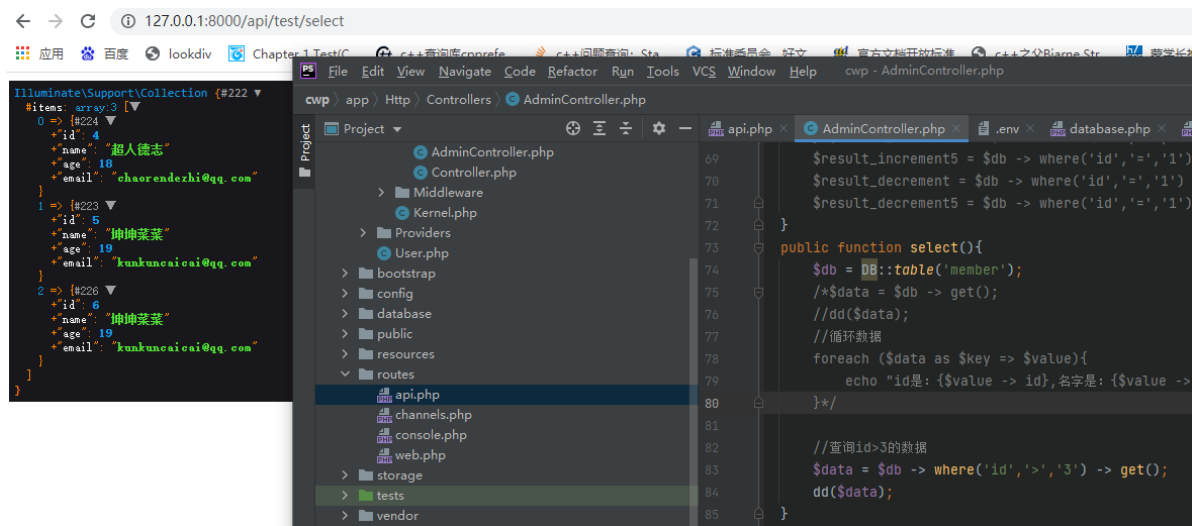
返回值是一个集合对象



循环遍历



get查询的结果每一行的记录是对象的形式，不是数组



where

where方法之后继续调用where方法

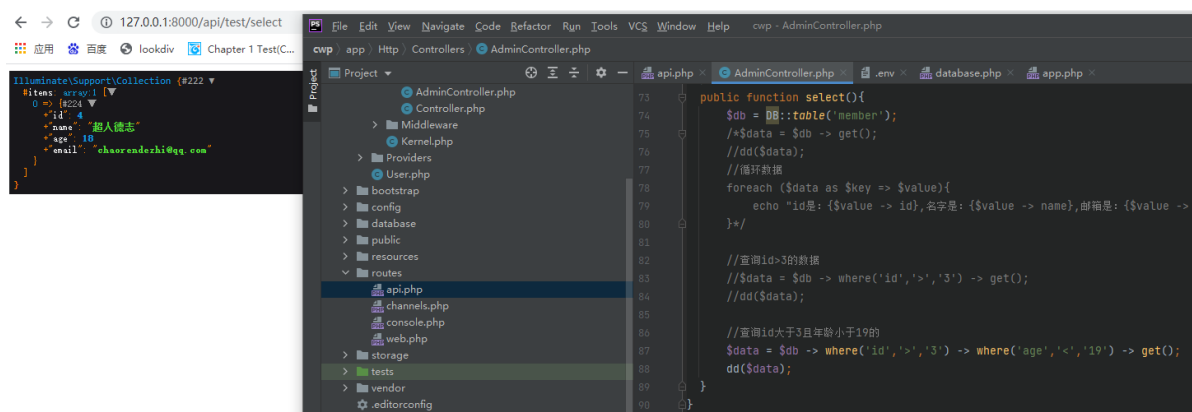
->where()->where()->where()..

这个语法是**并且(and)**关系语法

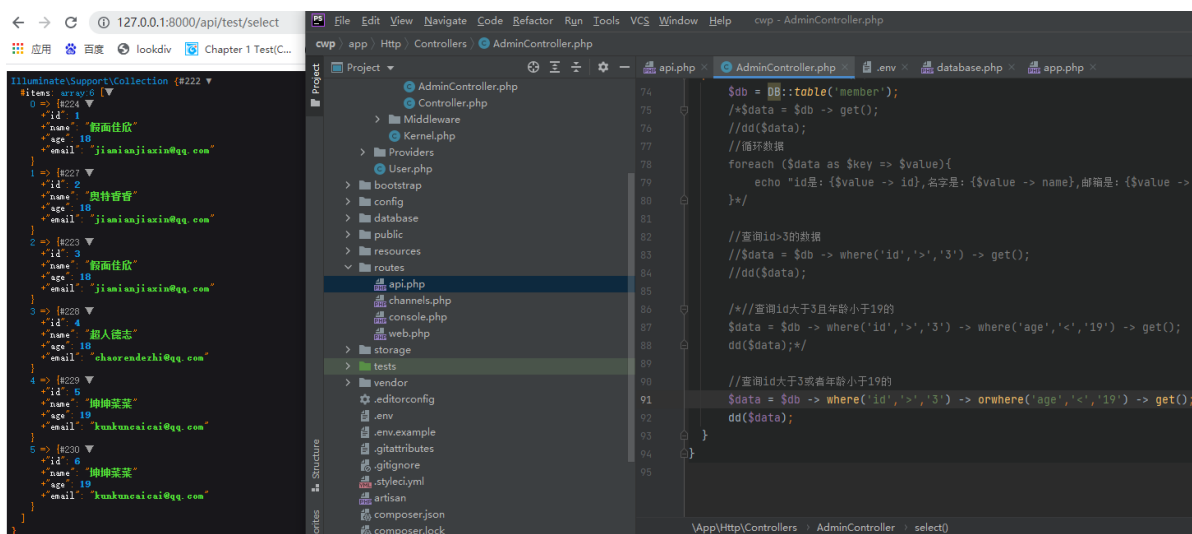
->where()->orWhere()->orWhere()..

这个语法是**或者(or)**关系语法

案例：并且 (and)



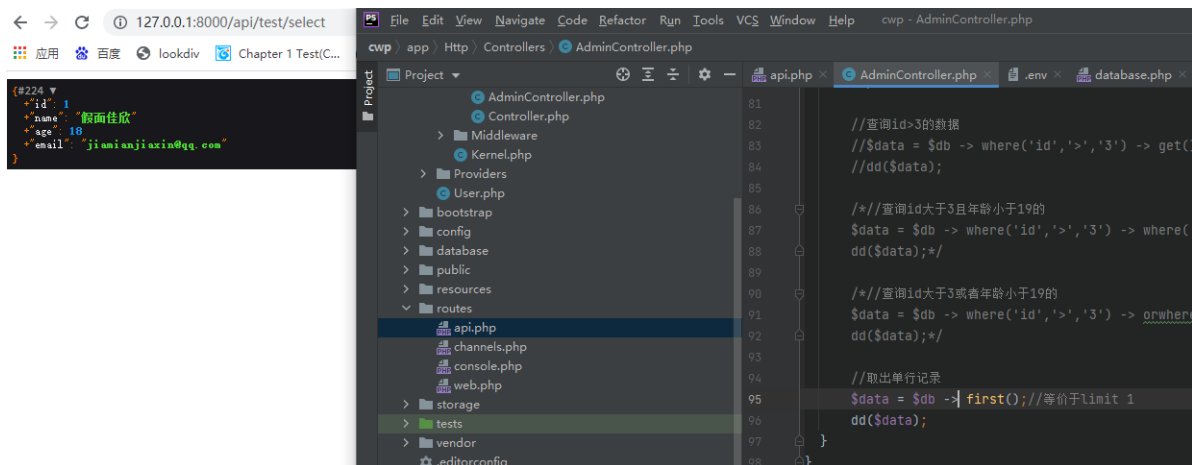
或者 (or)



取出单行数据

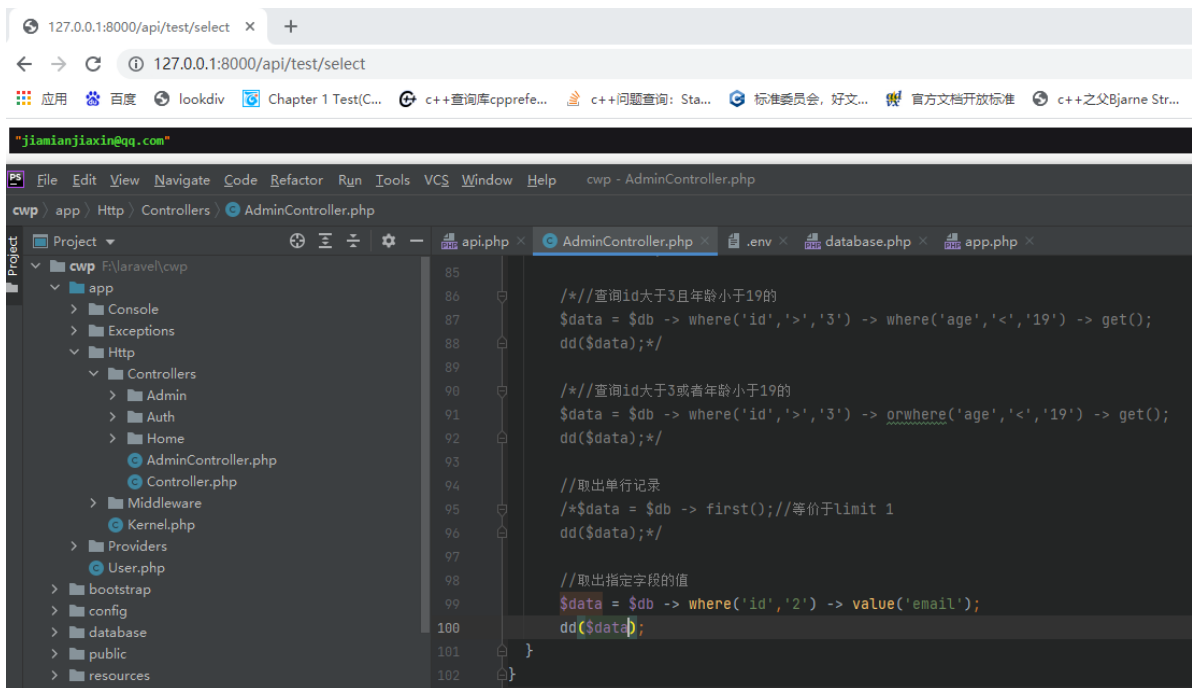
DB::table('member')->where('id','1')->first(); //返回值是一个对象

等价于limit 1



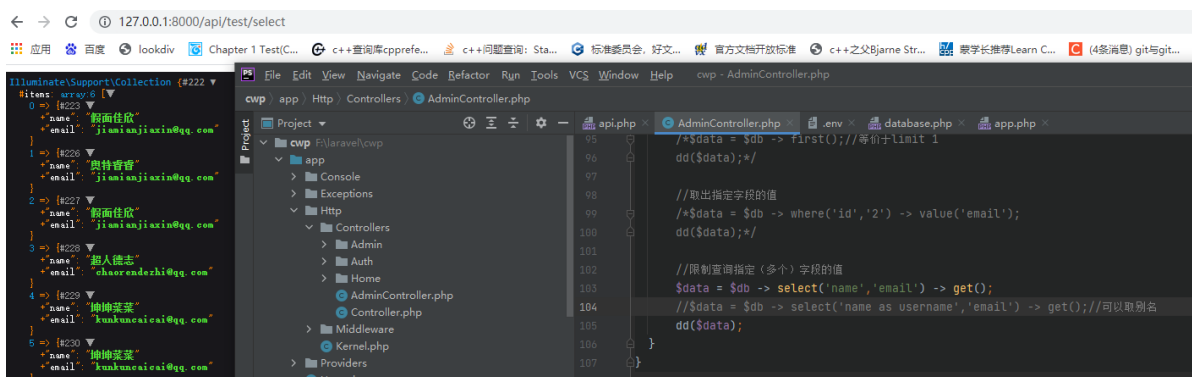
获取某个具体的值（一个字段）

格式：DB::table('member')->where('id','1')->value('name');



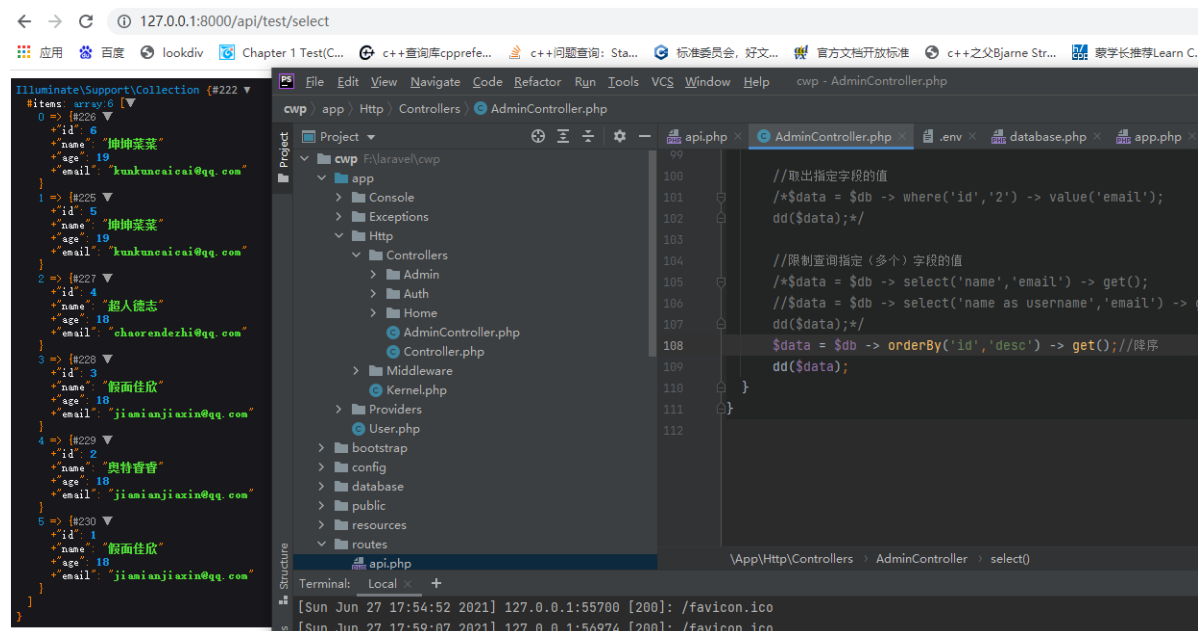
获取某些字段数据（多个字段）

格式：DB::table('member') -> select('name','email') -> get();



排序操作

DB::table('member')->orderBy('age','desc')->get();



```
127.0.0.1:8000/api/test/select

Illuminate\Support\Collection (#222)
items: array:6 [▼]
  0 => {#228 ▼
    +id: 6
    +name: "坤坤菜菜"
    +age: 19
    +email: "kunkuncaicai@qq.com"
  }
  1 => {#225 ▼
    +id: 5
    +name: "坤坤菜菜"
    +age: 19
    +email: "kunkuncaicai@qq.com"
  }
  2 => {#227 ▼
    +id: 4
    +name: "超人健志"
    +age: 18
    +email: "chaorenzhi@qq.com"
  }
  3 => {#228 ▼
    +id: 3
    +name: "假面佳欣"
    +age: 18
    +email: "jiamianjiaxin@qq.com"
  }
  4 => {#229 ▼
    +id: 2
    +name: "奥特曼香"
    +age: 18
    +email: "jiamianjiaxin@qq.com"
  }
  5 => {#230 ▼
    +id: 1
    +name: "假面佳欣"
    +age: 18
    +email: "jiamianjiaxin@qq.com"
  }
]

Project: cwp
  app
    Console
    Exceptions
    Http
      Controllers
        AdminController.php
        Controller.php
    Middleware
      Kernel.php
    Providers
      User.php
  bootstrap
  config
  database
  public
  resources
  routes
  api.php

AdminController.php
100 //取出指定字段的值
101 /*$data = $db -> where('id','2') -> value('email');
102 dd($data);*/
103
104 //限制查询指定 (多个) 字段的值
105 /*$data = $db -> select('name','email') -> get();
106 // $data = $db -> select('name as username','email') ->
107 dd($data);*/
108 $data = $db -> orderBy('id','desc') -> get(); //降序
109 dd($data);
110 }
111
112

Terminal: Local
[Sun Jun 27 17:54:52 2021] 127.0.0.1:55700 [200]: /favicon.ico
[Sun Jun 27 17:59:07 2021] 127.0.0.1:56974 [200]: /favicon.ico
```

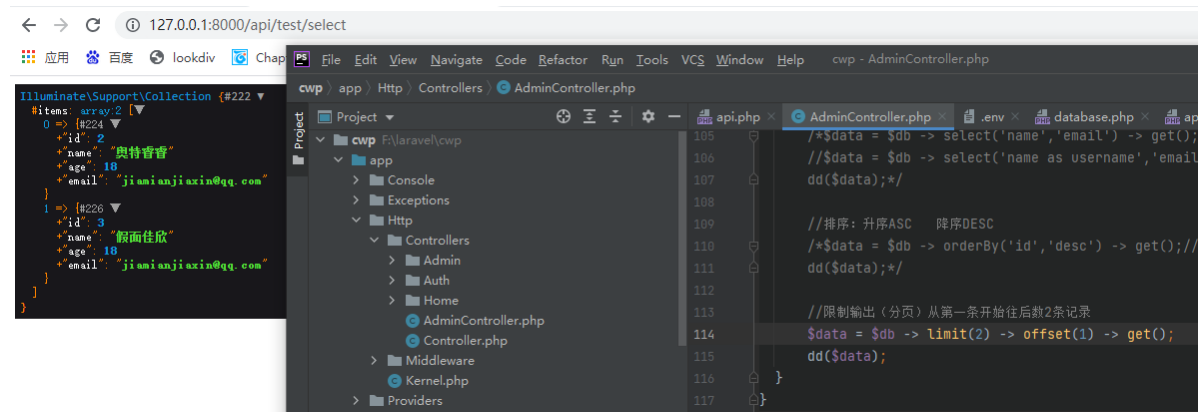
分页操作

格式: DB::table('member')->limit(3)->offset(2)->get();

Limit: 表示限制输出的条数

Offset: 从什么地方开始

等价于 limit 3,2的使用方法 (例子)



```
127.0.0.1:8000/api/test/select

Illuminate\Support\Collection (#222)
items: array:2 [▼]
  0 => {#224 ▼
    +id: 2
    +name: "奥特曼香"
    +age: 18
    +email: "jiamianjiaxin@qq.com"
  }
  1 => {#226 ▼
    +id: 3
    +name: "假面佳欣"
    +age: 18
    +email: "jiamianjiaxin@qq.com"
  }
]

Project: cwp
  app
    Console
    Exceptions
    Http
      Controllers
        AdminController.php
        Controller.php
    Middleware
      Kernel.php
    Providers
      User.php
  bootstrap
  config
  database
  public
  resources
  routes
  api.php

AdminController.php
105 /*$data = $db -> select('name','email') -> get();
106 // $data = $db -> select('name as username','email') ->
107 dd($data);*/
108
109 //排序: 升序ASC 降序DESC
110 /*$data = $db -> orderBy('id','desc') -> get();*/
111 dd($data);
112
113 //限制输出 (分页) 从第一条开始往后数2条记录
114 $data = $db -> limit(2) -> offset(1) -> get();
115 dd($data);
116 }
117
```

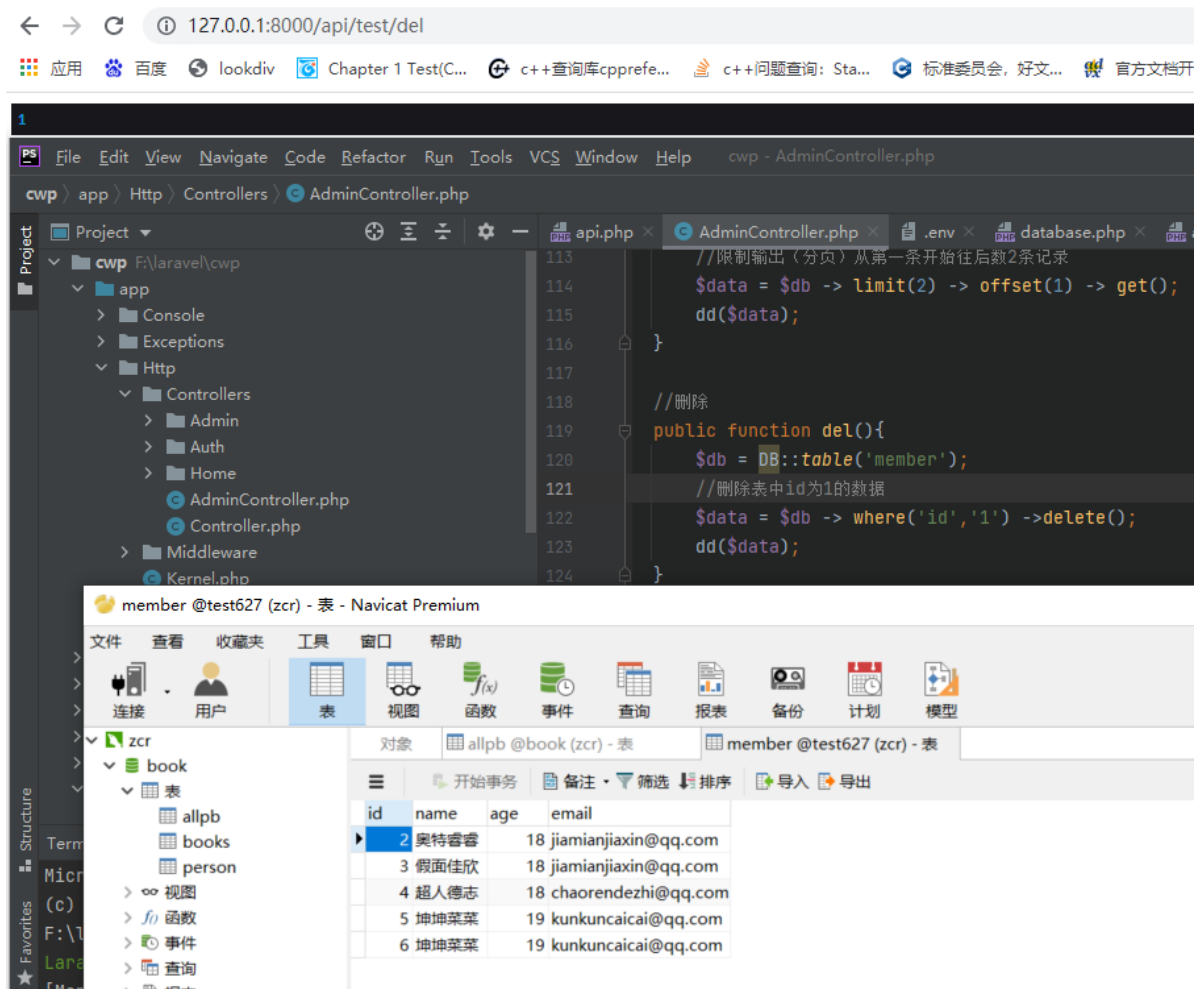
删除 (delete)

数据删除可以通过delete函数和truncate函数实现

Delete: 表示删除记录

Truncate 表示清空整个数据表;

DB::table('表名')->where('id','1')->delete();



返回值为受到影响行数

执行任意原生sql

- 执行任意的 **insert update delete** 的语句【影响记录数据的语句使用 **statement** 语法】
DB::statement("insert into member values(null, '');");
- 执行任意的 **select** 语句【不影响记录数据的语句使用 **select** 语法】
DB::select("select * from member");

问题

问题描述

1. laravel 取出单行数据时 first() 和 get() 有什么区别?
2. 如何跨多个表来保存信息? 例如登录信息

问题分析

1. <https://learnku.com/articles/46145> （暂定）
2. 通过session 来保存用户的登录信息！

问题解决

- 2.

```
session_start();//只要使用了session都需要在文件中添加此行代码
$id = $_post['id']//获取html传来的登录id信息
$_SESSION['id'] = $id;//通过session来保存id,然后在其他文件中就能通过将$_SESSION['id']赋
给其他变量就能获取全局的id了
```