

Laravel 5.5 入门教程

By IT崖柏图

Mail 973714522@qq.com

出自 布尔教育PHP高端教育培训

18章 投资功能

18.1 首页展示

路由: `Route::get('/', 'IndexController@index');`

```
use App\Pro;  
  
public function index(){  
    $pros = Pro::where('status',1)->take(3)->get();  
    return view('index',['pros'=>$pros]);  
}
```

view:

```
@foreach($pros as $p)  
<div class="pro_bor">  
    <ul class="clearfix pro_list_box">  
        <li class="w1">  
            <p class="fc_9 mb_5">预期年化 </p>  
            <p class="fc_orange airal"><em class="fs_28">{{ $p->rate }}</em>  
                <em class="fs_18">%</em>  
            </p>  
        </li>  
        <li class="w2">  
            <p class="fc_9 mb_5">期限</p>  
            <p class="fc_6"><em class="fs_18">{{ $p->hrange }}</em>个月</p>  
        </li>  
        <li class="w3">  
            <p class="fc_9 mb_5">投资金额 </p>  
            <p class="fc_6 airal fs_18">{{ $p->recive/100 }}</p>  
        </li>  
        <li class="w4">  
            <span class="b_jingdu b_jd">{{ sprintf('%d', 100 * $p->recive/$p->money) }}</span>{{ sprintf('%.2f', 100 * $p->recive/$p->money) }}%  
        </li>  
        <li class="w5">  
            <a class="btn btnSize_1 btn_orange" href="{{ url('pro', [$p->pid]) }}">立即投标</a>  
        </li>  
    </ul>  
</div>  
@endforeach
```

18.2 投标页展示

路由: `Route::get('pro/{pid}', 'ProController@pro');`

控制器:

```
public function pro($pid){  
    $pro = Pro::where(['pid'=>$pid, 'status'=>1])->first();  
    return view('lijitouzi',['pro'=>$pro]);  
}
```

view:

```
<ul class="clearfix td_all">
  <li class="fl td1">
    <h4 class="fc_9 pl_20">产品名称</h4>
    <p class="pl_20 fs_18 fc_6">
      <a href="#" title="{{ $pro->title }}">
        {{ $pro->title }}
      </a>
    </p>
  </li>
  <li class="fl td2">
    <h4 class="fc_9">借款年利率</h4>
    <p class="airal fs_24 fc_orange">
      {{ $pro->rate }}%
    </p>
  </li>
  <li class="fl td3">
    <h4 class="fc_9 pl_15">还款期限</h4>
    <p class="fs_18 pl_15">
      {{ $pro->hrange }}个月
    </p>
  </li>
  <li class="fl td4">
    <h4 class="fc_9">还款方式</h4>
    <p>按月付息 到期还本</p>
  </li>
  <li class="fl td5">
    <h4 class="fc_9">投资金额(元)</h4>
    <p><span class="airal fs_24 fc_orange">
      {{ ($pro->money - $pro->recive) / 100 }}
    </span>元</p>
  </li>
</ul>
```

18.3 投资表及迁移文件

投标表 :bids

字段	类型	说明
bid	primary	key 主键
uid	integer	用户 uid
pid	integer	项目 pid
title	string	项目名称
money	integer	投标金额
pubtime	integer	投标时间

创建迁移文件: php artisan make:migration create_table_bids --create=bids

```
// bids 表 迁移文件
public function up()
{
    Schema::create('bids', function (Blueprint $table) {
        $table->increments('bid');
        $table->integer('uid');
        $table->integer('pid');
        $table->string('title',50);
        $table->integer('money');
        $table->integer('pubtime');
    });
}
```

执行迁移命令: php artisan migrate

18.4 在线投标

view:

```
//resources/views/lijitouzi.blade.php
<form action="{url('touzi',[$pro->pid])}" method="post">
    {!! csrf_field() !!}
    <table width="100%" class="fc_6">
        <tbody>
            <tr>
                <th class="fs_16 fc_9" style="text-align:left;">投资金额: </th>
                <td style="padding-right:0;width:220px;">
                    <input style="width:140px;" class="fc_orange airal fs_18" name="money" value="{{($pro->money - $pro->recive) / 100}}">
                </td>
            </tr>
        </tbody>
    </table>
    <div class="fc_6 t_r b1 newstyle" style="padding-top:100px;">
        <input type="submit" id="confirmBut" value="确认购买" class="btn btnSize_1 btn_orange">
    </div>
</form>
```

路由: Route::post('touzi/{pid}','ProController@touzi');

生成Model: php artisan make:model Bid

```
//按照约定修改
class Bid extends Model
{
    protected $primaryKey = 'bid';
    public $timestamps = false;
}
```

写入数据到投资表:

```
// app/Http/Controllers/ProController.php
use App\Bid;
use Auth;

public function touzi(Request $request , $pid) {
    $pro = Pro::find($pid);
    $bid = new Bid();
    $user = Auth::user();
    if($req->money > ($pro->money-$pro->recive)){
        return "金额不能多于";
    }
    $bid->uid = $user->uid;
    $bid->pid = $pro->pid;
    $bid->title = $pro->title;
    $bid->money = $req->money * 100;
    $bid->pubtime = time();

    $bid->save(); // 写入投资信息
}
```

写入成功, 再次修改 touzi方法,
投资的钱, 不能大于项目剩余可投的钱,
一旦项目的钱, 等于已招标的钱, 修改状态为2, 还款中,
状态大于1的项目, 不能再投钱, 停止招标;

```
public function touzi(Request $req , $pid){
```

```

$pro = Pro::find($pid);
$bid = new Bid();
$user = Auth::user();

$bid->uid = $user->uid;
$bid->pid = $pro->pid;
$bid->title = $pro->title;
$bid->money = $req->money*100;//将用户投资的钱转化成分
$bid->pubtime = time();
$bid->save();//写入投资信息表

$pro->recive += $bid->money;//用户投资的钱我们需要把它写入项目表
$pro->save();

if($pro->recive == $pro->money){
    $this->touziDone($pid);
}
return '购买成功';
}

```

18.5 还款表与收益表

```

protected function touziDone($pro){
    // 1:修改状态为 2 ， 还款收益中
    $pro->status = 2;
    $pro->save();

    // 2: 为借款者，生成还款记录

    // 3:为投资者，生成收益记录
}

```

投标完成后，借款人每月 要还 "本 + 息", 投资人每天要收利息。
因此,我们重建2张表。

还款表 :hks

字段	类型	说明
hid	primary	key 主键
uid	integer	用户 uid
pid	integer	项目 pid
title	string	项目名称
amount	integer	每月还款金额
paydate	date	账单日
status	tinyint	是否已还

生成迁移文件：
php artisan make:migration create_table_hks --create=hks

```

//还款表 迁移文件
public function up()
{
    Schema::create('hks', function (Blueprint $table) {
        $table->increments('hid');
        $table->integer('uid');
        $table->integer('pid');
        $table->string('title');
    });
}

```

```

        $table->integer('amount');
        $table->date('paydate');
        $table->tinyinteger('status');
    });
}

```

执行迁移命令： php artisan migrate

18.6 生成月账单

**** 为借款者生成还款记录 ****

当投资完成的瞬间，由 projects 中的借款金额/借款周期/利率，算出借款人每月几号应还多少钱？

```

protected function touziDone($pid){
    // 1:修改状态为 2 ， 还款收益中
    $pro=Pro::find($pid);
    $pro->status = 2;
    $pro->save();

    // 2: 为借款者，生成还款记录
    //每月 应还 钱数：（本金 + 利息） 利率按百分比算，所以 除以100
    $amount = ($pro->money * $pro->rate / 1200 ) + ($pro->money/$pro->hrange);

    $row = ['uid'=>$pro->uid,'pid'=>$pro->pid,'title'=>$pro->title];
    $row['amount'] = $amount;
    $row['status'] = 0;

    //循环产生每月还款记录
    $today = date('Y-m-d',time()); //今天的日期
    for ($i=1; $i<=$pro->hrange ; $i++) {
        $paydate = date('Y-m-d',strtotime("+ $i months")); //下一个月的日期
        $row['paydate'] = $paydate;
        DB::table('hks')->insert($row);
    }

    // 3:为投资者，生成收益记录
}

```

18.7 生成预期收益

投资完成后，生成用户预期收入表，做定时任务，根据预期收益表 **** 每天 **** 给投资者产生利息收益，写入收益表；

预期收益表 tasks:

字段	类型	说明
tid	primary	key 主键
uid	integer	用户 uid
pid	integer	项目 pid
title	string	项目名称
amount	integer	每天应收的利息
enddate	date	收利息截止日

生成迁移文件： php artisan make:migration create_table_tasks --create=tasks

```

// tasks 表迁移文件
public function up()

```

```
{
    Schema::create('tasks', function (Blueprint $table) {
        $table->increments('tid');
        $table->integer('uid');
        $table->integer('pid');
        $table->string('title');
        $table->integer('amount');
        $table->date('enddate');
    });
}
```

执行迁移命令： php artisan migrate

```
protected function touziDone($pid) {
    ....
    ....
    // 3. 为投资者 ，生成收益打款任务
    $bids = Bid::where('pid' , $pid)->get(); //查询投资者信息
    $row = [];
    $row['pid'] = $pid;
    $row['title'] = $pro->title;
    //收益截止日期
    $row['enddate'] = date('Y-m-d' , strtotime("+ {$pro->hrange} months"));

    //循环为每位投资者生成 预期收益信息
    foreach($bids as $bid) {
        $row['uid'] = $bid->uid;
        $row['amount'] = $bid->money * $pro->rate / 36500;
        DB::table('tasks')->insert($row);
    }
}
```

18.8 生成收益

每天收益表 :grows

字段	类型	说明
gid	primary key	主键
uid	integer	用户 uid
pid	integer	项目 pid
title	string	项目名称
amount	integer	每天的利息
paytime	date	收益日期

```
// grows 表迁移文件
Schema::create('grows', function (Blueprint $table) {
    $table->increments('gid');
    $table->integer('uid');
    $table->integer('pid');
    $table->string('title',60);
    $table->integer('amount');
    $table->date('paytime');
});
```

生成控制器文件： php artisan make:controller GrowController --plain

```
// GrowController.php
public function run() {
```

```

$today = date('Y-m-d'); //先获取今天的日期

//在tasks表中，只要是大于今天日期的，都应该得到收益；
$tasks = DB::table('tasks')->where('enddate' , '>' , $today)->get();

//循环为每个人写入收益信息
foreach($tasks as $t) {
    $t->paytime = $today;
    $t = (array)$t;
    unset($t['tid']);
    unset($t['enddate']);
    DB::table('grows')->insert($t);
}
}

```

添加路由：Route::get('payrun','GrowController@run');

18.9 借款者 还款 账单

```

public function myzd() {
    $user = Auth::user();
    $hks = DB::table('hks')->where('uid',$user->uid)->get();
    return view('myzd' , ['hks'=>$hks]);
}

```

```

@foreach($hks as $h)
<tr>
    <td class="f">{{ $h->hid }}</td>
    <td class="tr"><span>{{ $h->title }}</span></td>
    <td class="tr">{{ $h->amount/100 }}</td>
    <td class="tr">{{ $h->paydate }}</td>
    <td class="tr">
        @if ($h->status == 0)
        <a href="#"> 立即还款 </a>
        @else
        已还
        @endif
    </td>
</tr>
@endforeach

```

18.10 我的投资

写路由：

```

public function mytz() {
    $user = Auth::user();
    $bids = Bid::where('bids.uid' , $user->uid)->whereIn('status' , [1,2])->join
('projects' , 'bids.pid' , '=' , 'projects.pid')->get(['bids.*' , 'projects.status']);
    //dd($bids);
    return view('mytz' , ['bids'=>$bids]);
}

//view
@foreach($bids as $b)
<tr>
    <td class="f">{{ $b->pid }}</td><td class="tr"><span>{{ $b->title }}</span></td>
    <td class="tr">{{ $b->money/100 }}</td>
    <td class="tr">{{ date('Y/m/d' , $b->pubtime) }}</td>
    <td class="tr">
        @if ($b->status == 1)
        招标中 , 快叫你的朋友来投资
        @elseif ($b->status == 2)
        收益中 , 发了
        @elseif ($b->status == 3)

```

```
        已结束
        @endif
    </td>
</tr>
@endforeach
```

18.11 我的收益

```
public function mysy() {
    $user = Auth::user();
    $grows = DB::table('grows')->where('uid' , $user->uid)->orderBy('gid' , 'desc')->get();
    return view('mysy' , ['grows'=>$grows]);
}
```

```
@foreach($grows as $g)
<tr>
    <td class="f">{{ $g->gid }}</td>
    <td class="tr"><span>{{ $g->title }}</span></td>
    <td class="tr"> {{ $g->amount/100 }} 元 </td>
    <td class="tr">{{ $g->paytime }}</td>
</tr>
@endforeach
```