<?php

namespace App\Modules\Callback\Controllers;

use Controller;

use View;

use Config;

use Input;

use Log;

use Cache,Lang,Redirect;

use App\Libraries\Cloudapi;

use App\Libraries\Curl;

class CallbackController extends BaseController {

/\*\*

\* Show the Conference

\*

\* @author yingchun

\*

\* @return \Illuminate\View\View

\*/

public function index()

{

$this->layout->content = View::make('callback::call');

}

public function getCall()

{

$countrycode=Config::get("callback::app.countrycode");

return View::make('callback::call')->with("countrycode",$countrycode);

}

public function postCall(){

$OssbssApi = new \App\Ossbss\OssbssApi();

$res = $OssbssApi->isAvailable();

if($res){

$input = Input::all();

Log::info("tel input all :".json\_encode($input));

$call = $this->getTelFormat($input);

if(!$call) return "My Number is error...";

$called = $this->getDestinationTelFormat($input);

if(!$called) return "Destination Number is error...";

$tag = md5(date("Y-m-d h:i:s").$called);

Cache::put($tag,$called,3600);

$result = $this->outCall($call,$tag);

if($result == 200)

{

return "call out successfully!please waiting...";

}

else

{

return "call out faild...";

}

}else{

return Lang::get('cpanel::message.recharge');

}

}

public function transfer(){

$cloudSession = json\_decode(file\_get\_contents("php://input"),true);

$tag = $cloudSession['session']['tag'];

Log::info("call back transfer to:".Cache::get($tag));

$cloudapi = new \App\Libraries\Cloudapi\Cloudapi();

$cloudapi->saySer("please wait,we are connecting your call!",array('voice'=>'en','attempts'=>1,'name'=>'wait'));

$cloudapi->transferSer(Cache::get($tag),array('timeout'=>60000,'name'=>'transfer','early\_media'=>'/tmp/future.mp3'));

Log::info("after transfer...");

$cloudapi->on(array('event'=>'hangup','next'=>'/hangup'));

$cloudapi->renderJSON();

Log::info("end....");

return "";

}

public function hangup(){

$result = json\_decode(file\_get\_contents("php://input"),true);

\App\Modules\Usages\Controllers\UsagesController::postCdrs($result);

}

protected function getTelFormat($input){

$mine="";

$myType = $input['myType'];

Log::info("my type:".$myType);

if("tel" == $myType){

$mine = $myType.":".$input['CountryCode'].$input['myNumber'];

}else if("sip" == $myType){

if(empty($input['mySipAccount'])) return false;

$base = Config::get('base::app');

Log::info("base :".json\_encode($base));

$mine = $myType.":".$input['mySipAccount']."@".$base["sip\_domain"];

}

Log::info("my tel is :".$mine);

return $mine;

}

protected function getDestinationTelFormat($input){

$destination="";

$destinationType = $input['destinationType'];

Log::info("dest type :".$destinationType);

if("tel" == $destinationType){

$destination = $destinationType.":".$input['destCountryCode'].$input['destinationNumber']; //new

}else if("sip" == $destinationType){

if(empty($input['destinationSipAccount'])) return false;

$base = Config::get('base::app');

$destination = $destinationType.":".$input['destinationSipAccount']."@".$base["sip\_domain"];

}

Log::info("dest tel:".$destination);

return $destination;

}

protected function outCall($call,$tag){

$outcall\_number = Config::get("callback::app.config.outcall\_number");

$outcall\_token = Config::get("callback::app.config.outcall\_token");

Log::info("call back out call to $call");

$kazoo = new \App\Libraries\Application\Kazoo();

$url = $kazoo->getUrl().$kazoo->outCall."?app\_number=".$outcall\_number."&call\_to\_number=".$call."&tag=".$tag;

Log::info("url is :".$url);

$curl = new \App\Libraries\Curl\mycurl($url);

$curl->setToken($outcall\_token);

$curl->setContentType("application/json; charset=utf-8");

$curl->createCurl();

Log::info("call back request out call url : $url");

Log::info("call back out call to $call result data:".$curl->\_\_tostring());

$status=$curl->getHttpStatus();

Log::info("http result status is ".$status);

return $status;

}

}