Acrescentando o SkipPresentar no ProjectRepositoru Eloquent

**<?php***/\*\*  
 \* Created by PhpStorm.  
 \* User: rafael.franco  
 \* Date: 22/03/2016  
 \* Time: 10:26  
 \*/***namespace** CodeProject\Repositories;  
  
  
**use** CodeProject\Entities\Project;  
**use** Prettus\Repository\Eloquent\BaseRepository;  
**use** CodeProject\Presenters\ProjectPresenter;  
  
  
**class** ProjectRepositoryEloquent **extends** BaseRepository **implements** ProjectRepository  
{  
 **public function** model()  
 {  
 **return** Project::*class*;  
 }  
  
 **public function** isOwner($projectID, $userId)  
 {  
 **if**(count($this->skipPresenter()->findWhere(['id'=>$projectID, 'owner\_id'=>$userId])))  
 {  
 **return true**;  
 }  
 **return false**;  
 }  
  
  
 **public function** hasMember($projectId, $memberId)  
 {  
 $project = $this->skipPresenter()->find($projectId);  
  
 **foreach**($project->members **as** $member) {  
 **if**($member->id == $memberId) {  
 **return true**;  
 }  
 }  
  
 **return false**;  
 }  
  
 **public function** presenter()  
 {  
 **return** ProjectPresenter::*class*;  
 }  
}

Nos Middleware

CheckProjectOwner

**<?php  
  
namespace** larang\Http\Middleware;  
  
**use** Closure;  
**use** CodeProject\Services\ProjectService;  
  
**class** CheckProjectOwner  
{  
  
 **protected** $service;  
  
 **public function** \_\_construct(ProjectService $service)  
 {  
 $this->service = $service;  
 }  
  
 **public function** handle($request, Closure $next)  
 {  
 $userId = \Authorizer::*getResourceOwnerId*();  
 $project\_id = $request->route('id') ? $request->route('id') : $request->route('project');  
 **if** (!$this->service->isOwner($userId, $project\_id)):  
 **return** ['error' => 'Access forbiden or inexistent project!'];  
 **endif**;  
 **return** $next($request);  
 }  
  
}

CheckProjectPermission

**<?php  
  
namespace** larang\Http\Middleware;  
  
**use** Closure;  
**use** CodeProject\Services\ProjectService;  
  
**class** CheckProjectPermission  
{  
  
 **protected** $service;  
  
 **public function** \_\_construct(ProjectService $service)  
 {  
 $this->service = $service;  
 }  
  
 **public function** handle($request, Closure $next)  
 {  
 $project\_id = $request->route('id') ? $request->route('id') : $request->route('project');  
 //dd($project\_id);  
 **if** (!$this->service->checkProjectPermitions($project\_id)):  
 **return** ['error' => 'Access forbiden or inexistent project!'];  
 **endif**;  
 **return** $next($request);  
 }  
  
}

Registrar no Kernel.php

*/\*\*  
 \* The application's route middleware.  
 \*  
 \** ***@var*** *array  
 \*/***protected** $routeMiddleware = [  
 'auth' => \CodeProject\Http\Middleware\Authenticate::*class*,  
 'auth.basic' => \Illuminate\Auth\Middleware\AuthenticateWithBasicAuth::*class*,  
 'guest' => \CodeProject\Http\Middleware\RedirectIfAuthenticated::*class*,  
 'csrf' => \CodeProject\Http\Middleware\VerifyCsrfToken::*class*,  
 'oauth' => \LucaDegasperi\OAuth2Server\Middleware\OAuthMiddleware::*class*,  
 'oauth-user' => \LucaDegasperi\OAuth2Server\Middleware\OAuthUserOwnerMiddleware::*class*,  
 'oauth-client' => \LucaDegasperi\OAuth2Server\Middleware\OAuthClientOwnerMiddleware::*class*,  
 'check-authorization-params' => \LucaDegasperi\OAuth2Server\Middleware\CheckAuthCodeRequestMiddleware::*class*,  
 'check.project.owner' => \CodeProject\Http\Middleware\CheckProjectOwner::*class*,  
 'check.project.permission' => \CodeProject\Http\Middleware\CheckProjectPermission::*class*,  
];

Alterar as Rotas

**<?php**/\*  
|--------------------------------------------------------------------------  
| Application Routes  
|--------------------------------------------------------------------------  
|  
| Here is where you can register all of the routes for an application.  
| It's a breeze. Simply tell Laravel the URIs it should respond to  
| and give it the controller to call when that URI is requested.  
|  
\*/  
  
Route::*get*('/', **function** () {  
 **return** view('app');  
});  
  
  
Route::*post*('oauth/access\_token', **function**() {  
 **return** Response::*json*(Authorizer::*issueAccessToken*());  
});  
  
  
Route::*group*(['middleware'=>'oauth'], **function**(){  
  
 Route::*resource*('client', 'ClientController', ['except' => ['create', 'edit']]);  
 Route::*resource*('project', 'ProjectController', ['except' => ['create', 'edit']]);  
// Route::group(['middleware'=> 'CheckProjectOwner'], function() {  
// Route::resource('project', 'ProjectController', ['except' => ['create', 'edit']]);  
// });  
  
 Route::*group*(['middleware'=>'check.project.permission', 'prefix'=>'project'], **function**() {  
  
 Route::*get*('{id}/note', 'ProjectNoteController@index');  
 Route::*post*('{id}/note', 'ProjectNoteController@store');  
 Route::*get*('{id}/note/{noteId}', 'ProjectNoteController@show');  
 Route::*put*('{id}/note/{noteId}', 'ProjectNoteController@update');  
 Route::*delete*('{id}/note/{noteId}', 'ProjectNoteController@destroy');  
  
 Route::*get*('{id}/task', 'ProjectTaskController@index');  
 Route::*post*('{id}/task', 'ProjectTaskController@store');  
 Route::*get*('{id}/task/{taskId}', 'ProjectTaskController@show');  
 Route::*put*('{id}/task/{taskId}', 'ProjectTaskController@update');  
 Route::*delete*('{id}/task/{taskId}', 'ProjectTaskController@destroy');  
  
 Route::*get*('{projectId}/member', 'ProjectMemberController@index');  
 Route::*post*('{projectId}/member', 'ProjectMemberController@store');  
 Route::*get*('{projectId}/member/{id}', 'ProjectMemberController@show');  
 Route::*delete*('{projectId}/member/{id}', 'ProjectMemberController@destroy');  
  
 Route::*get*('{projectId}/file', 'ProjectFileController@index');  
 Route::*post*('{projectId}/file', 'ProjectFileController@store');  
 Route::*get*('{projectId}/file/{id}', 'ProjectFileController@show');  
 Route::*get*('{projectId}/file/{id}/download', 'ProjectFileController@showFile');  
 Route::*delete*('{projectId}/file/{id}', 'ProjectFileController@destroy');  
 Route::*put*('{projectId}/file/{id}', 'ProjectFileController@update');  
 });  
  
 Route::*get*('user/authenticated', 'UserController@authenticated');  
  
});

Construtor ProjectController

**public function** \_\_construct(ProjectRepository $repository, ProjectService $service)  
{  
 $this->repository = $repository;  
 $this->service = $service;  
 $this->middleware('check.project.owner', ['except' => ['store', 'show', 'index']]);  
 $this->middleware('check.project.member', ['except' => ['store', 'update', 'destroy']]);  
}