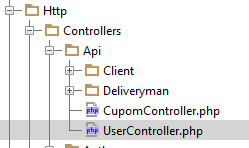
Ações do Capitulo

* Montar um menu de usuário
* E listagem de pedidos do cliente
* Utilizar um refresh na listagem

Criar Controller UserController.php em Api



**<?php  
  
namespace** CodeDelivery\Http\Controllers\Api;  
  
**use** CodeDelivery\Http\Controllers\Controller;  
**use** CodeDelivery\Http\Requests;  
**use** CodeDelivery\Repositories\UserRepository;  
**use** LucaDegasperi\OAuth2Server\Facades\Authorizer;  
  
  
**class** UserController **extends** Controller  
{  
  
 **private $userRepository**;  
  
 **public function** \_\_construct(UserRepository $userRepository)  
 {  
 $this->**userRepository** = $userRepository;  
 }  
  
 **public function** index()  
 {  
  
 }  
  
 **public function** authenticated()  
 {  
 $userId = Authorizer::*getResourceOwnerId*();  
 *//return $this->userRepository->with(['client'])->find($userId);* **return** $this->**userRepository**->skipPresenter(**false**)->find($userId);  
 }  
  
  
}

Chamar na Route

Route::*get*(**'authenticated '**, **'Api\UserController@authenticated'**);

UserRepositoryEloquent

**<?php  
  
namespace** CodeDelivery\Repositories;  
  
**use** Prettus\Repository\Eloquent\BaseRepository;  
**use** Prettus\Repository\Criteria\RequestCriteria;  
**use** CodeDelivery\Repositories\UserRepository;  
**use** CodeDelivery\Models\User;  
**use** CodeDelivery\Validators\UserValidator;  
  
*/\*\*  
 \* Class UserRepositoryEloquent  
 \** ***@package*** *namespace CodeDelivery\Repositories;  
 \*/***class** UserRepositoryEloquent **extends** BaseRepository **implements** UserRepository  
{  
 */\*\*  
 \* Specify Model class name  
 \*  
 \** ***@return*** *string  
 \*/* **protected $skipPresenter** = **true**;  
  
 **public function** model()  
 {  
 **return** User::***class***;  
 }  
  
 **public function** getDeliverymen()  
 {  
 **return** $this->**model**->where([**'role'**=>**'deliveryman'**])->lists(**'name'**, **'id'**);  
 }  
  
 */\*\*  
 \* Boot up the repository, pushing criteria  
 \*/* **public function** boot()  
 {  
 $this->pushCriteria(app(RequestCriteria::***class***));  
 }  
  
 **public function** presenter()  
 {  
 **return** \CodeDelivery\Presenters\UserPresenter::***class***;  
 }  
}

UserTransformer

**<?php  
  
namespace** CodeDelivery\Transformers;  
  
**use** League\Fractal\TransformerAbstract;  
**use** CodeDelivery\Models\User;  
  
*/\*\*  
 \* Class UserTransformer  
 \** ***@package*** *namespace CodeDelivery\Transformers;  
 \*/***class** UserTransformer **extends** TransformerAbstract  
{  
  
 **protected $availableIncludes** = [**'client'**];  
 */\*\*  
 \* Transform the \User entity  
 \** ***@param*** *\User $model  
 \*  
 \** ***@return*** *array  
 \*/* **public function** transform(User $model)  
 {  
 **return** [  
 **'id'** => (int) $model->**id**,  
 **'name'** => $model->**name**,  
 **'email'** => $model->**email**,  
 **'role'** => $model->**role** ];  
 }  
  
 **public function** includeClient(User $model)  
 {  
 **return** $this->item($model->**client**, **new** ClientTransformer());  
 }  
}

Retornando os Dados

http://localhost:8000/api/authenticated?include=client