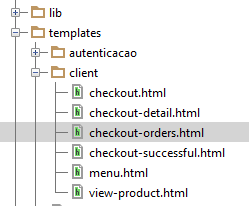
Mostrar as informações do pedido do cliente

Adicionando nomes de produtos no transformes OrderTransformer

**<?php  
  
namespace** CodeDelivery\Transformers;  
  
**use** Illuminate\Database\Eloquent\Collection;  
**use** League\Fractal\TransformerAbstract;  
**use** CodeDelivery\Models\Order;  
  
*/\*\*  
 \* Class OrderTransformer  
 \** ***@package*** *namespace CodeDelivery\Transformers;  
 \*/***class** OrderTransformer **extends** TransformerAbstract  
{  
 **protected $availableIncludes** = [**'cupom'**, **'items'**, **'client'**, **'deliveryman'**];  
 *// tenho que passar os includes por paramentros atraves do GET  
 //protected $availableIncludes = [];  
 /\*\*  
 \* Transform the \Order entity  
 \** ***@param*** *\Order $model  
 \*  
 \** ***@return*** *array  
 \*/* **public function** transform(Order $model)  
 {  
  
 **return** [  
 **'id'** => (int) $model->**id**,  
 **'total'** => $model->**total**,  
 **'product\_names'** => $this->getArrayProductNames($model->**items**),  
 **'status'** => \CodeDelivery\Http\Helpers\formatStatusOrder($model->**status**),  
 */\* place your other model properties here \*/* **'created\_at'** => $model->**created\_at**,  
 **'updated\_at'** => $model->**updated\_at** ];  
 }  
  
 **protected function** getArrayProductNames(Collection $items)  
 {  
 $names = [];  
 **foreach**($items **as** $item)  
 {  
 $names[] = $item->**product**->**name**;  
 }  
 **return** $names;  
 }  
  
 **public function** includeClient(Order $model)  
 {  
 **return** $this->item($model->**client**, **new** ClientTransformer());  
 }  
  
 **public function** includeDeliveryman(Order $model)  
 {  
 **if**(!$model->**deliveryman**)  
 **return null**;  
  
 **return** $this->item($model->**deliveryman**, **new** DeliverymanTransformer());  
 }  
  
 *//Many to One - CUmpo  
 /\*****TODO MEDOTO PARA SERIALZAR O RELACIONAMENTO****\*/* **public function** includeCupom(Order $model)  
 {  
 *//VALIDANDO O CUPOM - PQ O CUPOM PODE OU NAO PODE EXISTIR* **if**(!$model->**cupom**)  
 **return null**;  
  
 **return** $this->item($model->**cupom**, **new** CupomTransformer());  
 }  
  
 *//One To Many - Items* **public function** includeItems(Order $model)  
 {  
 **return** $this->collection($model->**items**, **new** OrderItemTransformer());  
 }  
}

View checkout-orders.html



<**ion-view**>  
 <**ion-nav-title**>Meus Pedidos</**ion-nav-title**>  
 <**ion-content class="padding has-header"**>  
 <**ion-list**>  
 <**ion-item ng-repeat="o in orders"**>  
 <**div class="row"**>  
 <**div class="col"**>  
 <**h2**> PED#{{o.id}} em {{ o.created\_at.date|limitTo:10|date:'dd/MM/yyyy'}}</**h2**>  
 <**p**>{{o.product\_names | limitTo:2 }}</**p**>  
 <**p**>{{o.status}}</**p**>  
 <**i class="ion ion-ios-cart"**> {{o.product\_names.length}} Items</**i**>  
 </**div**>  
 <**div class="col text-right"**>  
 <**i class="ion ion-social-usd"**> {{o.total}}</**i**>  
 </**div**>  
 </**div**>  
  
 </**ion-item**>  
 </**ion-list**>  
 </**ion-content**>  
</**ion-view**>