**1.Route Model Binding**

At first create a make:controller **PostCommentController**

Then make a request **php artisan make:request StoreComment**

in web.php create a new route

**Route::resource('posts.comments', 'PostCommentController')->only(['store']);**

Go to **StoreComment.php**

**public function rules()**

**{**

**return [**

**//**

**'content'=>'required:min:5'**

**];**

**}**

**make public function authorize(){**

**return true;**

**}**

**in PostCommentController.php**

**go and write public function \_\_construct(){**

**$this->middleware('auth')->only(['store']);**

**}**

**We need to go to \_form.blade.php**

**<form method="POST" action="{{ route('posts.comments.store', ['post' => $posts->id]) }}">**

**Comment.php**

**public static function boot()**

{

parent::boot();

// static::addGlobalScope(new LatestScope);

static::creating(function (Comment $comment) {

Cache::forget("blog-post-{$comment->blog\_post\_id}");

});

}

**2.EagerLoading Nested Relationships**

**Refactored code in PostsController.php**

return BlogPost::with('comment','tags','user','comment.user')->findorFail($id);

Refactored code **PostTagController.php**

**public function index($tag)**

**{**

**$tag= Tag::findorFail($tag);**

**return view('posts.index',**

**['posts'=>$tag->blogPost()**

**->latest()**

**->with('user')**

**->with('tags')**

**->get()**

**// 'most\_commented'=>[],**

**// 'mostActive'=>[],**

**// 'mostActiveLastMonth'=>[],**

**]**

**);**

**}**

**3.Converting repeated query to Query scopes**

**Refactor code in BlogPost.php**

**public function scopelatestwithRelations(Builder $query ){**

**return $query->latest()**

**->withCount('comment')**

**->with('user')**

**->with('tags');**

**}**

**in PostsController**

**return view(**

**'posts.index',**

**[**

**'posts'=>BlogPost::latestwithRelations()->get(), ]**

**);**

**in PostsTagController**

**['posts'=>$tag->blogPost()->latestwithRelations()->get()**

**]**