Começando deste ponto? Você pode fazer o [*download*](https://s3.amazonaws.com/caelum-online-public/940-angular-parte3/stages/13-alurapic.zip) completo do projeto do capítulo anterior e continuar seus estudos a partir deste capítulo. Será necessário abrir seu terminal, entrar dentro da pasta *alurapic* e executar o comando ***npm install*** para baixar novamente todas as dependências da aplicação.

Este curso é a continuação do curso [Angular parte 2: Autenticação, Forms e lazy loading](https://www.alura.com.br/curso-online-angular-autenticacao).

Template do componente:

<footer class="mt-5" \*ngIf="(user$ | async) as user">

<div class="fixed-bottom bg-white p-1">

<div class="container">

<div class="row text-center">

<div class="col-6">

<a [routerLink]="['']">

<i class="fa fa-home fa-2x"></i>

</a>

</div>

<div class="col-6">

<a [routerLink]="['p', 'add']">

<i class="fa fa-plus-circle fa-2x"></i>

</a>

</div>

</div>

</div>

</div>

</footer>

Formulário, template do componente:

<div class="container">

<form class="row">

<div class="col-md-6 text-center">

<input type="file" accept="image/\*">

</div>

<div class="col-md-6">

<div class="form-group">

<textarea

class="form-control form-control-sm"

placeholder="photo description"></textarea>

</div>

<div class="form-group">

<label class="text-muted">

Allow comments

<input type="checkbox">

</label>

</div>

<button type="submit" class="btn btn-primary btn-block">

Upload

</button>

<a class="btn btn-secondary btn-block">Cancel</a>

</div>

</form>

</div>

import { Component, OnInit } from "@angular/core";

import { AuthService } from "../../core/auth/auth.service";

import { FormGroup, FormBuilder } from "@angular/forms";

import { Validators } from "@angular/forms";

@Component({

selector: 'ap-photo-form',

templateUrl: './photo-form.component.html',

})

export class PhotoFormComponent implements OnInit {

photoForm: FormGroup;

constructor(private formBuilder: FormBuilder) { }

ngOnInit(): void {

this.photoForm = this.formBuilder.group({

file: ['', Validators.required],

description: ['', Validators.maxLength(300)],

allowComments: [true]

});

}

}

<div class="container">

<form [formGroup]="photoForm" class="row">

<div class="col-md-6 text-center">

<input formControlName="file" type="file" accept="image/\*">

<ap-vmessage

text="Please, select a photo"

\*ngIf="photoForm.get('file').errors?.required">

</ap-vmessage>

</div>

<div class="col-md-6">

<div class="form-group">

<textarea

formControlName="description"

class="form-control form-control-sm"

placeholder="photo description"></textarea>

<ap-vmessage

text="Maximun size allowed is 300"

\*ngIf="photoForm.get('description').errors?.maxlength">

</ap-vmessage>

</div>

<div class="form-group">

<label class="text-muted">

Allow comments

<input formControlName="allowComments" type="checkbox">

</label>

</div>

<button

[disabled]="photoForm.invalid"

type="submit" class="btn btn-primary btn-block">

Upload

</button>

<a class="btn btn-secondary btn-block">Cancel</a>

</div>

</form>

</div>

import { HttpClient, HttpParams } from '@angular/common/http';

import { Injectable } from '@angular/core';

import { Photo } from "./photo";

const API = 'http://localhost:3000';

@Injectable({ providedIn: 'root' })

export class PhotoService {

constructor(private http: HttpClient) {}

listFromUser(userName: string) {

return this.http

.get<Photo[]>(API + '/' + userName + '/photos');

}

listFromUserPaginated(userName: string, page: number) {

const params = new HttpParams()

.append('page', page.toString());

return this.http

.get<Photo[]>(API + '/' + userName + '/photos', { params });

}

upload(formData: FormData) {

return this.http.post(

API + '/photos/upload', formData

);

}

}

import { Component, OnInit } from "@angular/core";

import { AuthService } from "../../core/auth/auth.service";

import { FormGroup, FormBuilder } from "@angular/forms";

import { Validators } from "@angular/forms";

@Component({

selector: 'ap-photo-form',

templateUrl: './photo-form.component.html',

})

export class PhotoFormComponent implements OnInit {

photoForm: FormGroup;

constructor(private formBuilder: FormBuilder) { }

ngOnInit(): void {

this.photoForm = this.formBuilder.group({

file: ['', Validators.required],

description: ['', Validators.maxLength(300)],

allowComments: [true]

});

}

upload() {

console.log('chamou upload');

}

}

<div class="container" (submit)="upload()">

<form [formGroup]="photoForm" class="row">

import { Component, Input, OnInit } from '@angular/core'

const cloud = 'http://localhost:3000/imgs/';

@Component({

selector: 'ap-photo',

templateUrl: './photo.component.html'

})

export class PhotoComponent {

@Input() description = '';

private \_url = '';

@Input() set url(url: string) {

if(url && url.startsWith('data')) {

this.\_url = url;

} else {

this.\_url = cloud + url;

}

}

get url() {

return this.\_url;

}

}

Começando deste ponto? Você pode fazer o [*download*](https://s3.amazonaws.com/caelum-online-public/940-angular-parte3/stages/14-alurapic.zip) completo do projeto do capítulo anterior e continuar seus estudos a partir deste capítulo. Será necessário abrir seu terminal, entrar dentro da pasta *alurapic* e executar o comando ***npm install*** para baixar novamente todas as dependências da aplicação.

Template do botão

<button type="button" (click)="fileInput.click()" class="btn btn-primary">

<i class="fa fa-image fa-4x align-middle"></i>

</button>

Começando deste ponto? Você pode fazer o [*download*](https://s3.amazonaws.com/caelum-online-public/940-angular-parte3/stages/15-alurapic.zip) completo do projeto do capítulo anterior e continuar seus estudos a partir deste capítulo. Será necessário abrir seu terminal, entrar dentro da pasta *alurapic* e executar o comando ***npm install*** para baixar novamente todas as dependências da aplicação.

import { Component } from "@angular/core";

@Component({

templateUrl: './photo-detail.component.html',

})

export class PhotoDetailComponent implements OnInit {

constructor() {}

}

import { NgModule } from '@angular/core';

import { Routes, RouterModule } from '@angular/router';

import { PhotoListComponent } from './photos/photo-list/photo-list.component';

import { PhotoFormComponent } from './photos/photo-form/photo-form.component';

import { NotFoundComponent } from './errors/not-found/not-found.component';

import { PhotoListResolver } from './photos/photo-list/photo-list.resolver';

import { AuthGuard } from './core/auth/auth.guard';

import { PhotoDetailComponent } from './photos/photo-detail/photo-detail.component';

const routes: Routes = [

{

path: '',

pathMatch: 'full',

redirectTo: 'home'

},

{

path: 'home',

loadChildren: './home/home.module#HomeModule'

},

{

path: 'user/:userName',

component: PhotoListComponent,

resolve: {

photos: PhotoListResolver

}

},

{

path: 'p/add',

component: PhotoFormComponent,

canActivate: [AuthGuard]

},

{

path: 'p/:photoId', // nova rota

component: PhotoDetailComponent

},

{

path: '\*\*',

component: NotFoundComponent

}

];

@NgModule({

imports: [

RouterModule.forRoot(routes, { useHash: true } )

],

exports: [ RouterModule ]

})

export class AppRoutingModule { }

<div class="bg-white border">

<div class="row">

<div class="col-lg-8">

<ap-photo [url]="photo?.url" [description]="photo?.description"></ap-photo>

</div>

<div class="col-lg-4">

<small>

<p class="text-left break-word">{{photo?.description}}</p>

<hr>

</small>

<div class="mt-4">

<form>

<div class="input-group">

<textarea class="form-control"></textarea>

<div class="input-group-append">

<button class="btn btn-primary pull-left">Publish</button>

</div>

</div>

</form>

</div>

</div>

</div>

</div>

Começando deste ponto? Você pode fazer o [*download*](https://s3.amazonaws.com/caelum-online-public/940-angular-parte3/stages/16-alurapic.zip) completo do projeto do capítulo anterior e continuar seus estudos a partir deste capítulo. Será necessário abrir seu terminal, entrar dentro da pasta *alurapic* e executar o comando ***npm install*** para baixar novamente todas as dependências da aplicação.

<ul class="list-unstyled">

<li \*ngFor="let comment of comments">

<a

class="text-dark font-weight-bold mr-1"

[routerLink]="['/user', comment.userName]">

{{ comment.userName }}

</a>

<em>{{ comment.date | date:'dd-MM-yyyy HH:mm' }}</em>

<p class="break-word mb-2">{{ comment.text }}</p>

</li>

</ul>

<ap-vmessage

text="Maximun size allowed is 300"

\*ngIf="commentForm.get('comment').errors?.maxlength">

</ap-vmessage>

import { NgModule } from "@angular/core";

import { PhotoDetailComponent } from "./photo-detail.component";

import { RouterModule } from "@angular/router";

import { PhotoModule } from "../photo/photo.module";

import { CommonModule } from "@angular/common";

import { PhotoCommentsComponent } from "./photo-comments/photo-comments.component";

import { ReactiveFormsModule } from "@angular/forms";

@NgModule({

declarations: [

PhotoDetailComponent,

PhotoCommentsComponent

],

imports: [

CommonModule,

RouterModule,

PhotoModule,

ReactiveFormsModule

]

})

export class PhotoDetailModule { }

import { Component } from "@angular/core";

import { ActivatedRouteSnapshot, ActivatedRoute } from "@angular/router";

import { OnInit } from "@angular/core";

import { Photo } from "../photo/photo";

import { PhotoService } from "../photo/photo.service";

import { Observable } from "rxjs";

import { FormGroup, FormBuilder } from "@angular/forms";

import { Validators } from "@angular/forms";

import { PhotoCommentsComponent } from "./photo-comments/photo-comments.component";

import { PhotoComment } from "../photo/photo-comment";

@Component({

templateUrl: './photo-detail.component.html',

})

export class PhotoDetailComponent implements OnInit {

photo: Photo;

comments: PhotoComment[];

commentForm: FormGroup;

constructor(

private route: ActivatedRoute,

private photoService: PhotoService,

private formBuilder: FormBuilder

) {}

ngOnInit(): void {

const photoId = this.route.snapshot.params.photoId;

this.photoService

.findById(photoId)

.subscribe(photo => this.photo = photo);

this.photoService

.getComments(photoId)

.subscribe(comments => this.comments = comments);

this.commentForm = this.formBuilder.group({

comment: ['', Validators.compose([

Validators.required,

Validators.maxLength(300)

])]

});

}

Temos o seguinte trecho de código:

save() {

const comment = this.commentForm.get('comment').value as string;

this.comments$ = this.photoService

.addComment(this.photoId, comment)

.pipe(switchMap(() => this.photoService.getComments(this.photoId)))

.pipe(tap(() => {

this.commentForm.reset();

}));

}

O operador switchMap cancela o Observable anterior passando o fluxo para um novo Observable, garantindo assim que a emissão tenha apenas o valor emitido pelo Observable retornado por switchMap.

Começando deste ponto? Você pode fazer o [*download*](https://s3.amazonaws.com/caelum-online-public/940-angular-parte3/stages/17-alurapic.zip) completo do projeto do capítulo anterior e continuar seus estudos a partir deste capítulo. Será necessário abrir seu terminal, entrar dentro da pasta *alurapic* e executar o comando ***npm install*** para baixar novamente todas as dependências da aplicação.

Começando deste ponto? Você pode fazer o [*download*](https://s3.amazonaws.com/caelum-online-public/940-angular-parte3/stages/17-alurapic.zip) completo do projeto do capítulo anterior e continuar seus estudos a partir deste capítulo. Será necessário abrir seu terminal, entrar dentro da pasta *alurapic* e executar o comando ***npm install*** para baixar novamente todas as dependências da aplicação.

Template

<div class="text-left mb-4">

<i class="fa fa-comment-o fa-2x mr-2 ml-2"></i>{{ photo.comments }}

<i class="fa fa-trash-o fa-2x pull-right"></i>

</div>

Começando deste ponto? Você pode fazer o [*download*](https://s3.amazonaws.com/caelum-online-public/940-angular-parte3/stages/19-alurapic.zip) completo do projeto do capítulo anterior e continuar seus estudos a partir deste capítulo. Será necessário abrir seu terminal, entrar dentro da pasta *alurapic* e executar o comando ***npm install*** para baixar novamente todas as dependências da aplicação.

Template dos likes:

<i class="fa fa-heart-o fa-2x mr-2 pull-left"> {{ photo.likes }} </i>

like(photoId: number) {

return this.http.post(

API + '/photos/' + photoId + '/like', {}, { observe: 'response'}

)

.pipe(map(res => true))

.pipe(catchError(err => {

return err.status == '304' ? of(false): throwError(err);

}));

}

like(photo: Photo) {

this.photoService

.like(photo.id)

.subscribe(liked => {

if(liked) {

this.photo$ = this.photoService.findById(photo.id);

}

});

}

Perdeu o projeto completo do curso? Não se preocupe, você pode baixá-lo [aqui](https://s3.amazonaws.com/caelum-online-public/940-angular-parte3/stages/20-alurapic.zip).