



ANGULAR 2 FUNDAMENTALS

December 13, 2017

PART 16

DELETING AND IMAGE USING LARAVEL

➤ Open ImagesController.php

```
public function destroy($id)
{
    $image = Image::find($id);

    if(!$image){
        return Response::json([
            'error' => [
                'message' => "Cannot find the image."
            ]
        ], 404);
    }
}
```

DELETING AND IMAGE USING LARAVEL

- Continue on ImagesController.php

```
$image->delete();

$message = 'Your image has been deleted successfully';

$response = Response::json([
    'message' => $message,
], 201);

return $response;
}
```


DELETE AN IMAGE USING ANGULAR

- `ng g c admin-image-delete`
- Open `admin.routes.ts` add a new route:

```
export const adminRoutes: Routes = [  
  { path: '', component: DashboardComponent },  
  { path: 'images', component: AdminImageListComponent },  
  { path: 'images/create', component: AdminImageCreateComponent },  
  { path: 'images/edit/:id', component: AdminImageEditComponent },  
  { path: 'images/delete/:id', component: AdminImageDeleteComponent },  
];
```

- Remember to import `AdminImageDelete` component:

```
import { AdminImageDeleteComponent } from './admin-image-delete/admin-image-delete.component';
```

ADMIN-IMAGE-DELETE.COMPONENT.TS

- Create 2 variables:

```
id: any;  
params: any;
```

- Use ActivatedRoute, ImageService and Router

```
constructor(private activatedRoute: ActivatedRoute, private imageService: ImageService,  
private router: Router) { }
```

- Don't forget to import:

```
import {ActivatedRoute, Router} from '@angular/router';  
import {ImageService} from '../../services/image.service';
```

CONTINUED

➤ Update ngOnInit:

```
ngOnInit() {  
  this.params = this.activatedRoute.params.subscribe(params => this.id = params['id']);  
  
  this.imageService.deleteImage(this.id).subscribe(  
    data => {  
      console.log(data);  
      this.router.navigate(['/admin/images']);  
    },  
    error => console.log(<any>error));  
}
```

➤ Recap:

- On the code above we get the id through this.params
- Once we have it we use the image service deleteImage method to send delete request to backend

CONTINUED

- Additionally to prevent memory leaks again we should unsubscribe

```
ngOnDestroy() {  
  this.params.unsubscribe();  
}
```

- Find:

```
export class AdminImageDeleteComponent implements OnInit {
```

- Update:

```
export class AdminImageDeleteComponent implements OnInit, OnDestroy {
```

- Importing OnDestroy

```
import {Component, OnInit, OnDestroy} from '@angular/core';
```


ADMIN-IMAGE-DELETE.COMPONENT.TS SUMMARIZED

```
import {Component, OnDestroy, OnInit} from '@angular/core';
import {ActivatedRoute, Router} from '@angular/router';
import {ImageService} from '../../../services/image.service';

@Component({
  selector: 'ng-admin-image-delete',
  templateUrl: './admin-image-delete.component.html',
  styleUrls: ['./admin-image-delete.component.css']
})
export class AdminImageDeleteComponent implements OnInit, OnDestroy {
  id: any;
  params: any;

  constructor(private activatedRoute: ActivatedRoute, private imageService: ImageService, private router: Router) { }

  ngOnInit() {
    this.params = this.activatedRoute.params.subscribe(params => this.id = params['id']);
    this.imageService.deleteImage(this.id).subscribe(
      data => {
        console.log(data);
        this.router.navigate(['/admin/images']);
      },
      error => console.log(<any>error));
  }

  ngOnDestroy() {
    this.params.unsubscribe();
  }
}
```

LASTLY OPEN IMAGE.SERVICE.TS

- Add deleteImage method:

```
deleteImage(id: String): Observable<Image[]> {  
  const apiUrl = 'http://angularbook.app/api/v1/images';  
  const url = `${apiUrl}/${id}`;  
  return this.http.delete(url)  
    .map((response) => response)  
    .catch((error: any) => Observable.throw(error.json().error || {message: 'Server Error'}));  
}
```

- Try it out: <http://localhost:4200/admin/images> and delete an image
- Congrats on a full working crud!

END OF PART 16