*import* {Component, *OnInit*, *ViewChild*} from '@angular/core';  
*import* {FormBuilder, Validators} *from* '@angular/forms';  
*import* {ToolService} *from* '../tool.service';  
*import* {*Tool*} *from* '../tool';  
*import* {Router} *from* '@angular/router';  
*import* {*error*} *from* '@angular/compiler/src/util';  
  
  
@Component({  
 selector: 'app-new-tool',  
 templateUrl: './new-tool.component.html',  
 styleUrls: ['./new-tool.component.css']  
})  
*export class* NewToolComponent *implements OnInit* {  
  
 fileNames: *string*[] = [];  
 @ViewChild('fileUpload', {*static*: *false*}) fileUpload: ElementRef; files = [];  
  
 newToolForm = *this*.fb.group({  
 title: ['', Validators.*required*],  
 photos: [''],  
 description: ['', Validators.*required*],  
 category: ['', Validators.*required*],  
 price: ['', Validators.*required*],  
 quantity: ['', Validators.*required*],  
 });  
  
 *constructor*(*private* fb: FormBuilder, *private* toolService: ToolService, *private* route: Router) {  
 }  
  
 ngOnInit(): *void* {  
 }  
  
 onSubmit() {  
  
 *// this.toolService.newTool(this.tool).subscribe(() => this.route.navigate(['home']), error1 => console.log(error1), () => console.log('New tool created sucessfully'));  
 const* formData = *new* FormData();  
 formData.append('title', *this*.newToolForm.get('title').value);  
 formData.append('photos', *this*.newToolForm.get('photos').value);  
 formData.append('description', *this*.newToolForm.get('description').value);  
 formData.append('category', *this*.newToolForm.get('category').value);  
 formData.append('price', *this*.newToolForm.get('price').value);  
 formData.append('quantity', *this*.newToolForm.get('quantity').value);  
  
 console.log(*this*.newToolForm.get('title').value);  
 console.log(*this*.newToolForm.get('photos').value);  
  
 *this*.toolService.newTool(formData).subscribe(  
 () => *this*.route.navigate(['home']),  
 error1 => console.log(error1),  
 () => console.log('New tool created sucessfully'));  
 }  
  
 resetForm() {  
 *this*.newToolForm.reset();  
 }  
  
 fileEvent(event?: *Event*) {  
 *const* amount = (event.target *as HTMLInputElement*).files.length;  
 *for* (*let* i = 0; i < amount; i++) {  
 *this*.fileNames.push((event.target *as HTMLInputElement*).files[i].name);  
 *const* file = (event.target *as HTMLInputElement*).files[i];  
 *this*.newToolForm.patchValue({  
 photos: file  
 });  
 *this*.newToolForm.get('photos').updateValueAndValidity();  
 }  
 }  
  
}