```java
ackage com.example.demo.Controller;

import com.example.demo.Modele.Api;
import com.example.demo.Modele.Category;
import com.example.demo.Service.CategoryService;
import lombok.RequiredArgsConstructor;
import org.jetbrains.annotations.NotNull;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.http.ResponseEntity;
import org.springframework.validation.Errors;
import org.springframework.web.bind.annotation.*;

import javax.validation.Valid;
import java.util.ArrayList;

@RestController
@RequestMapping("Api/v1/Category")
@RequiredArgsConstructor
public class CategoryController {

    private final ArrayList<Category> category = new ArrayList<>();
    @Autowired
    private final CategoryService categoryService;


    @GetMapping(".get")
    public ResponseEntity<ArrayList<Category>> getCategory(){
        return ResponseEntity.status(200).body(category);
    }
```

```java
    @PostMapping(".add")

    public  ResponseEntity addCategory(@RequestBody @Valid Category category, @NotNull Errors
errors){

        if (errors.hasErrors()){

            return ResponseEntity.status(400).body(new
Api(errors.getFieldError().getDefaultMessage(),400));

        }


        categoryService.addCategoey(category);

        return ResponseEntity.status(200).body(new Api("New Category added",200));


    }


    @PutMapping(".updat")

    public ResponseEntity updateCategory(@RequestBody @Valid Category category, Errors errors) {

        if (errors.hasErrors()) {

            return ResponseEntity.status(400).body(new Api(errors.getFieldError().getDefaultMessage(),
400));

        }

        Boolean isCategoryAdd = categoryService.updateCategory(category);

        if (! isCategoryAdd){

            return ResponseEntity.status(200).body(new Api("New Category update", 200));

        }

        return ResponseEntity.status(400).body(new Api("Category not found",400));


    }


    @DeleteMapping(".delet")
```

```java
    public ResponseEntity deletCategory(@PathVariable String id, String categoryid){
        if(categoryService.deletCategory(id, categoryid)){
            return ResponseEntity.status(200).body(new Api("Category deleted",200));
        }
        return ResponseEntity.status(400).body(new Api("Category not found",400));


    }


}
```