

```
package rest;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.http.HttpStatus;
import org.springframework.web.bind.annotation.DeleteMapping;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.PutMapping;
import org.springframework.web.bind.annotation.RequestBody;
import org.springframework.web.bind.annotation.RequestParam;
import org.springframework.web.bind.annotation.RestController;
import org.springframework.web.server.ResponseStatusException;

@RestController
public class CatController {
    @Autowired
    private CatsRepo categoryRepo;

    @GetMapping("/cats")
    public List<Category> getAll() {
        return categoryRepo.findAll();
    }

    @GetMapping("/cats/desc/{desc}")
    public List<Category> getAllByDesc(@PathVariable("desc") String desc) {
        return categoryRepo.findByDescriptionContaining(desc);
    }

    @GetMapping("/cats/{code}")
    public Category getCategoryByCode(
        @PathVariable("code") String code) {
        var cat = categoryRepo.findById(code);
        if (cat.isPresent())
            return cat.get();
        else
            throw new ResponseStatusException(HttpStatus.NOT_FOUND,
                "Category Code Not Found");
    }

    @DeleteMapping("/cats/{code}")
    public void deleteCategoryByCode(@PathVariable("code") String code) {
        var cat = categoryRepo.findById(code);
        if (cat.isPresent())
            categoryRepo.deleteById(code);
        else
            throw new ResponseStatusException(HttpStatus.NOT_FOUND, "Category Code Not Found");
    }

    @PostMapping("/cats")
    public Category addCategory(@RequestBody Category category) {
        try {
            categoryRepo.save(category);
            return category;
        } catch (Exception ex) {
            throw new ResponseStatusException(HttpStatus.BAD_REQUEST, "Invalid data!");
        }
    }

    @PutMapping("/cats/{code}")
    public Category updateCategoryDesc(@PathVariable("code") String code,
```

```
        @RequestParam("desc") String desc) {  
    if (desc.length() == 0)  
        throw new ResponseStatusException(HttpStatus.BAD_REQUEST, "Invalid Description");  
  
    var cat = categoryRepo.findById(code);  
    if (cat.isPresent()) {  
        var category = cat.get();  
        category.setDescription(desc);  
        categoryRepo.save(category);  
        return category;  
    } else  
        throw new ResponseStatusException(HttpStatus.NOT_FOUND, "Category Code Not Found");  
    }  
}
```