

```
1 package com.crud.tasks;
2
3 import com.crud.tasks.domain.TaskDto;
4 import org.springframework.boot.SpringApplication;
5 import org.springframework.boot.autoconfigure.SpringBootApplication;
6
7 @SpringBootApplication
8 public class TasksApplication {
9
10     public static void main(String[] args) {
11
12         SpringApplication.run(TasksApplication.class, args);
13     }
14
15 }
16
```

```
1 package com.crud.tasks.domain;
2
3 import lombok.AllArgsConstructor;
4 import lombok.Getter;
5 import lombok.NoArgsConstructor;
6
7 import javax.persistence.Column;
8 import javax.persistence.Entity;
9 import javax.persistence.GeneratedValue;
10 import javax.persistence.Id;
11
12 @NoArgsConstructor
13 @AllArgsConstructor
14 @Getter
15 @Entity(name = "tasks")
16 public class Task {
17
18     @Id
19     @GeneratedValue
20     private Long id;
21
22     @Column(name = "name")
23     private String title;
24
25     @Column(name = "description")
26     private String content;
27
28 }
29
```

```
1 package com.crud.tasks.domain;
2
3 import lombok.AllArgsConstructor;
4 import lombok.Getter;
5
6 @Getter
7 @AllArgsConstructor
8 public class TaskDto {
9
10     private Long id;
11     private String title;
12     private String content;
13
14 }
15
```

```
1 package com.crud.tasks.mapper;
2
3 import com.crud.tasks.domain.Task;
4 import com.crud.tasks.domain.TaskDto;
5 import org.springframework.stereotype.Service;
6
7 import java.util.List;
8
9 @Service
10 public class TaskMapper {
11
12     public Task mapToTask(final TaskDto taskDto) {
13         return new Task(
14             taskDto.getId(),
15             taskDto.getTitle(),
16             taskDto.getContent()
17         );
18     }
19
20     public TaskDto mapToTaskDto(final Task task) {
21         return new TaskDto(
22             task.getId(),
23             task.getTitle(),
24             task.getContent()
25         );
26     }
27
28     public List<TaskDto> mapToTaskDtoList(final List<Task> taskList){
29         return taskList.stream()
30             .map(this::mapToTaskDto)
31             .toList();
32     }
33 }
34
```

```
1 package com.crud.tasks.service;
2
3 import com.crud.tasks.controller.TaskNotFoundException;
4 import com.crud.tasks.domain.Task;
5 import com.crud.tasks.repository.TaskRepository;
6 import lombok.RequiredArgsConstructor;
7
8 import org.springframework.stereotype.Service;
9
10 import java.util.List;
11 import java.util.Optional;
12
13 @Service
14 @RequiredArgsConstructor
15 public class DbService {
16     private final TaskRepository repository;
17
18     public List<Task> getAllTasks() {
19         return repository.findAll();
20     }
21
22     public Task getTask(final Long taskId) throws TaskNotFoundException {
23         return repository.findById(taskId).orElseThrow(TaskNotFoundException::new);
24     }
25
26     public Task saveTask(final Task task) {
27         return repository.save(task);
28     }
29
30     public void deleteById(final Long taskId) throws TaskNotFoundException {
31         getTask(taskId);
32         repository.deleteById(taskId);
33     }
34 }
35
```

```
1 package com.crud.tasks.controller;
2
3 import com.crud.tasks.domain.Task;
4 import com.crud.tasks.domain.TaskDto;
5 import com.crud.tasks.mapper.TaskMapper;
6 import com.crud.tasks.service.DbService;
7 import lombok.RequiredArgsConstructor;
8 import org.springframework.http.HttpStatus;
9 import org.springframework.http.MediaType;
10 import org.springframework.http.ResponseEntity;
11 import org.springframework.web.bind.annotation.*;
12
13 import java.awt.*;
14 import java.util.ArrayList;
15 import java.util.List;
16
17 @RestController
18 @RequestMapping("/v1/tasks")
19 @RequiredArgsConstructor
20 @CrossOrigin("*")
21 public class TaskController {
22
23     public final DbService service;
24     public final TaskMapper taskMapper;
25
26     @GetMapping
27     public ResponseEntity<List<TaskDto>> getTasks() {
28         List<Task> tasks = service.getAllTasks();
29         List<TaskDto> taskDtoList = taskMapper.mapToTaskDtoList(tasks);
30         return ResponseEntity.ok(taskDtoList);
31     }
32
33     @GetMapping(value = "{taskId}")
34     public ResponseEntity<TaskDto> getTask(@PathVariable Long taskId) throws TaskNotFoundException {
35         Task task = service.getTask(taskId);
36         TaskDto taskDto = taskMapper.mapToTaskDto(task);
37         return ResponseEntity.ok(taskDto);
38     }
39
40     @DeleteMapping(value = "{taskId}")
41     public ResponseEntity<Void> deleteTask(@PathVariable Long taskId) throws TaskNotFoundException {
42         service.deleteById(taskId);
43         return ResponseEntity.ok().build();
44     }
45
46     @PutMapping
47     public ResponseEntity<TaskDto> updateTask(@RequestBody TaskDto taskDto) {
48         Task task = taskMapper.mapToTask(taskDto);
49         Task savedTask = service.saveTask(task);
50         return ResponseEntity.ok(taskMapper.mapToTaskDto(savedTask));
51     }
52
53     @PostMapping(consumes = MediaType.APPLICATION_JSON_VALUE)
54     public ResponseEntity<Void> CreateTask(@RequestBody TaskDto taskDto) {
55         Task task = taskMapper.mapToTask(taskDto);
56         service.saveTask(task);
57         return ResponseEntity.ok().build();
58     }
59 }
```

```
1 package com.crud.tasks.controller;
2
3 public class TaskNotFoundException extends Exception {
4 }
5
```

```
1 package com.crud.tasks.controller;
2
3 import org.springframework.http.HttpStatus;
4 import org.springframework.http.ResponseEntity;
5 import org.springframework.web.bind.annotation.ControllerAdvice;
6 import org.springframework.web.bind.annotation.ExceptionHandler;
7 import org.springframework.web.servlet.mvc.method.annotation.ResponseEntityExceptionHandler;
8
9 @ControllerAdvice
10 public class GlobalExceptionHandler extends ResponseEntityExceptionHandler {
11
12     @ExceptionHandler(TaskNotFoundException.class)
13     public ResponseEntity<Object> handleTaskNotFoundException(TaskNotFoundException exception) {
14         return new ResponseEntity<>("Task with given id doesn't exist", HttpStatus.BAD_REQUEST);
15     }
16 }
17
```



```
1 package com.crud.tasks.repository;
2
3 import com.crud.tasks.domain.Task;
4 import org.springframework.data.repository.CrudRepository;
5
6 import org.springframework.stereotype.Repository;
7
8 import java.util.List;
9 import java.util.Optional;
10
11 @Repository
12 public interface TaskRepository extends CrudRepository<Task, Long> {
13     List<Task> findAll();
14     Optional<Task> findById(Long taskId);
15     Task save(Task task);
16 }
17
```