# Swagger/OpenAPI Annotations for Task Scheduler API

## 📘 Spring Boot Swagger Configuration

Add the following dependency in `pom.xml` to enable Swagger UI using SpringDoc:

<dependency>  
 <groupId>org.springdoc</groupId>  
 <artifactId>springdoc-openapi-ui</artifactId>  
 <version>1.6.15</version>  
</dependency>

This will expose API docs at:  
- Swagger UI: `http://localhost:8080/swagger-ui.html`  
- OpenAPI JSON: `http://localhost:8080/v3/api-docs`

## 🧩 Annotated API Example

Here’s how your TaskController can be documented with OpenAPI annotations:

@RestController  
@RequestMapping("/tasks")  
@Tag(name = "Task API", description = "Endpoints for task scheduling and status tracking")  
public class TaskController {  
  
 @Operation(summary = "Submit a new task")  
 @ApiResponses(value = {  
 @ApiResponse(responseCode = "200", description = "Task created"),  
 @ApiResponse(responseCode = "400", description = "Invalid input")  
 })  
 @PostMapping  
 public ResponseEntity<Task> submitTask(@Valid @RequestBody Task task) {  
 return ResponseEntity.ok(taskService.save(task));  
 }  
  
 @Operation(summary = "Get task by tenant and ID")  
 @ApiResponses(value = {  
 @ApiResponse(responseCode = "200", description = "Task found"),  
 @ApiResponse(responseCode = "404", description = "Task not found")  
 })  
 @GetMapping("/{tenantId}/{taskId}")  
 public ResponseEntity<Task> getTask(@PathVariable String tenantId, @PathVariable String taskId) {  
 Task task = taskService.findById(tenantId, taskId);  
 return task != null ? ResponseEntity.ok(task) : ResponseEntity.notFound().build();  
 }  
}

## 📌 Summary

By adding Swagger annotations and validations, your API becomes self-documenting, user-friendly, and compliant with OpenAPI standards — ensuring better developer usability and aligning with MCA project guidelines.