```
package com.example.rwtool.services;

import com.example.rwtool.dto.ReportRequest;
import com.example.rwtool.dto.ReportResponse;
import com.example.rwtool.dto.ReportTypeResponse;
import com.example.rwtool.entity.Report;
import com.example.rwtool.entity.ReportType;
import com.example.rwtool.repo.ReportRepository;
import com.example.rwtool.repo.ReportTypeRepository;
import org.springframework.stereotype.Service;

import java.util.List;
import java.util.UUID;

@Service
public class ReportService {

    private final ReportRepository reportRepo;
    private final ReportTypeRepository reportTypeRepo;

    public ReportService(ReportRepository reportRepo,
ReportTypeRepository reportTypeRepo) {
        this.reportRepo = reportRepo;
        this.reportTypeRepo = reportTypeRepo;
    }

    public List<ReportResponse> getAllReports() {
        return
reportRepo.findAll().stream().map(this::mapToResponse).toList();
    }

    public ReportResponse addReport(ReportRequest request) {
        ReportType reportType =
reportTypeRepo.findById(request.getReportTypeId())
                .orElseThrow(() -> new RuntimeException("ReportType not
found"));

        Report report = new Report();
        report.setFileName(request.getFileName());
        report.setReportType(reportType);
        report.setFileStoragePath(request.getFileStoragePath());
        report.setGeneratedDate(request.getGeneratedDate());
        report.setUploadedBy(request.getUploadedBy());

        Report saved = reportRepo.save(report);
        return mapToResponse(saved);
    }

    public ReportResponse updateReport(UUID id, ReportRequest request) {
        Report report = reportRepo.findById(id)
                .orElseThrow(() -> new RuntimeException("Report not
found"));
```

```
        ReportType reportType =
reportTypeRepo.findById(request.getReportTypeId())
                .orElseThrow(() -> new RuntimeException("ReportType not
found"));

        report.setFileName(request.getFileName());
        report.setReportType(reportType);
        report.setFileStoragePath(request.getFileStoragePath());
        report.setGeneratedDate(request.getGeneratedDate());
        report.setUploadedBy(request.getUploadedBy());

        Report updated = reportRepo.save(report);
        return mapToResponse(updated);
    }

    public void deleteReport(UUID id) {
        reportRepo.deleteById(id);
    }

    private ReportResponse mapToResponse(Report report) {
        ReportType type = report.getReportType();
        ReportTypeResponse typeResponse =
                new ReportTypeResponse(type.getReportTypeId(),
type.getName(), type.getSourcePath());

        return new ReportResponse(report.getReportId(),
                report.getFileName(),
                report.getFileStoragePath(),
                report.getGeneratedDate(),
                report.getUploadedBy(),
                typeResponse);
    }
}

package com.example.rwtool.controller;

import com.example.rwtool.dto.ReportRequest;
import com.example.rwtool.dto.ReportResponse;
import com.example.rwtool.services.ReportService;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.*;

import java.util.List;
import java.util.UUID;

@RestController
@RequestMapping("/api/reports")
@CrossOrigin(origins = "*")
public class ReportController {

    private final ReportService service;

    public ReportController(ReportService service) {
        this.service = service;
    }

    @GetMapping
    public List<ReportResponse> getAllReports() {
        return service.getAllReports();
    }

    @PostMapping
    public ReportResponse createReport(@RequestBody ReportRequest req) {
        return service.addReport(req);
    }

    @PutMapping("/{id}")
    public ReportResponse updateReport(@PathVariable UUID id,
@RequestBody ReportRequest req) {
        return service.updateReport(id, req);
    }

    @DeleteMapping("/{id}")
    public ResponseEntity<Void> deleteReport(@PathVariable UUID id) {
        service.deleteReport(id);
        return ResponseEntity.noContent().build();
    }
}
```