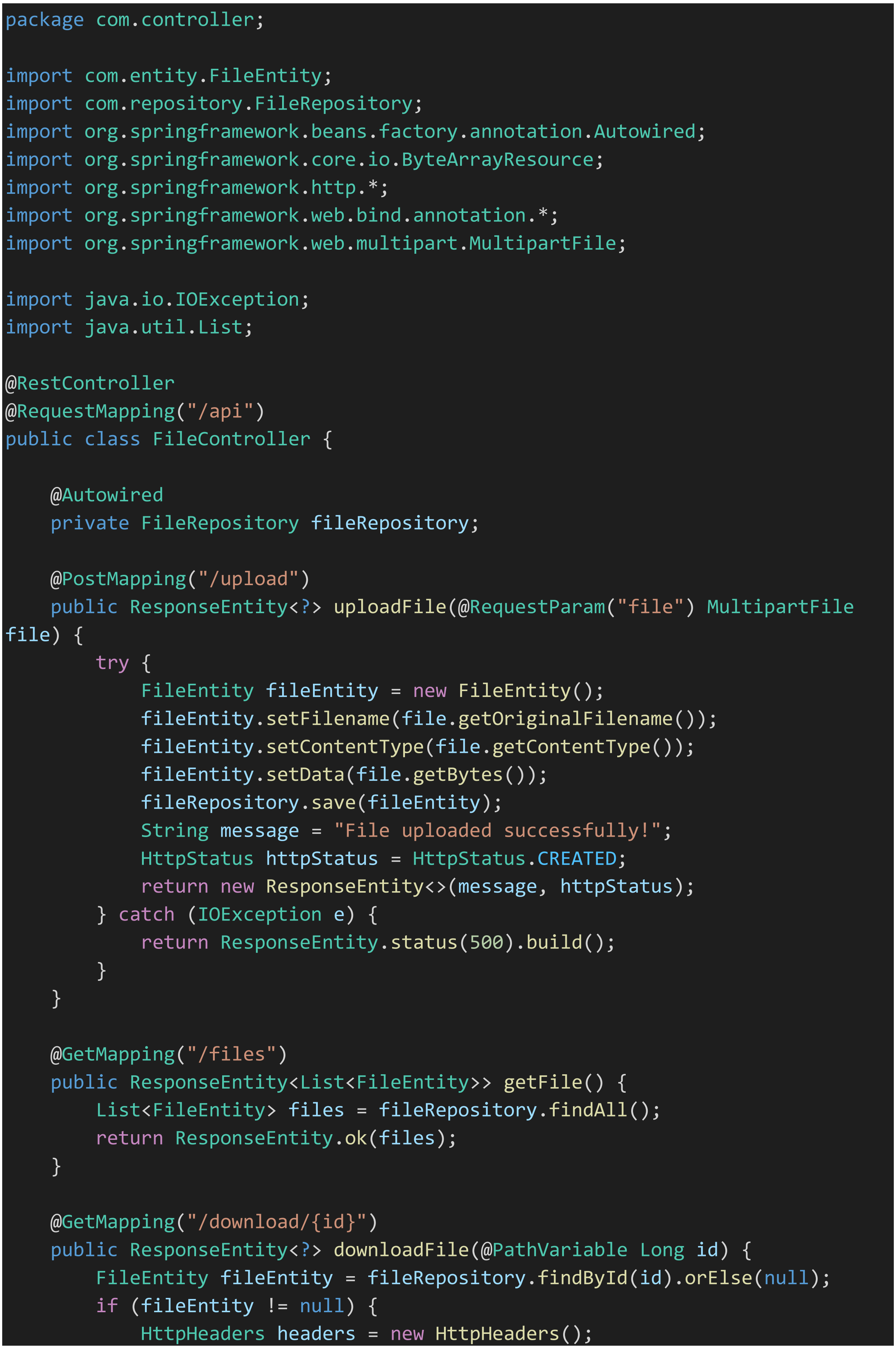

```
package com.controller;

import com.entity.FileEntity;
import com.repository.FileRepository;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.core.io.ByteArrayResource;
import org.springframework.http.*;
import org.springframework.web.bind.annotation.*;
import org.springframework.web.multipart.MultipartFile;

import java.io.IOException;
import java.util.List;

@RestController
@RequestMapping("/api")
public class FileController {

    @Autowired
    private FileRepository fileRepository;

    @PostMapping("/upload")
    public ResponseEntity<?> uploadFile(@RequestParam("file") MultipartFile 
file) {
        try {
            FileEntity fileEntity = new FileEntity();
            fileEntity.setFilename(file.getOriginalFilename());
            fileEntity.setContentType(file.getContentType());
            fileEntity.setData(file.getBytes());
            fileRepository.save(fileEntity);
            String message = "File uploaded successfully!";
            HttpStatus httpStatus = HttpStatus.CREATED;
            return new ResponseEntity<>(message, httpStatus);
        } catch (IOException e) {
            return ResponseEntity.status(500).build();
        }
    }

    @GetMapping("/files")
    public ResponseEntity<List<FileEntity>> getFile() {
        List<FileEntity> files = fileRepository.findAll();
        return ResponseEntity.ok(files);
    }

    @GetMapping("/download/{id}")
    public ResponseEntity<?> downloadFile(@PathVariable Long id) {
        FileEntity fileEntity = fileRepository.findById(id).orElse(null);
        if (fileEntity != null) {
            HttpHeaders headers = new HttpHeaders();
            headers.setContentType(MediaType.parseMediaType(fileEntity.getContentType()));
            headers.setContentDisposition(ContentDisposition.attachment().filename(fileEntity.getFilename()).build());
            ByteArrayResource resource = new ByteArrayResource(fileEntity.getData());
            return ResponseEntity.ok().headers(headers).body(resource);
        } else {
            return ResponseEntity.notFound().build();
        }
    }

}
```