

Endpoints para upload e download em uma API Spring Boot.

Vá até sua entidade mapeada em seu projeto, crie um campo "**byte []**", esse campo será responsável por armazenar o arquivo. Esse campo deve estar com anotado com "**@JsonIgnore**", para evitarmos devolver um vetor de bytes em um JSON. Além disso, não esqueça de mapear com "**@Column (length = tamanho_aqui)**" para evitar subir imagens extremamente grandes.

Exemplo:

```
@JsonIgnore // ignoramos no JSON pois não faz sentido retornar um vetor de bytes num JSON!
@Column(length = 50 * 1024 * 1024) // Equivalente a 50 Mega Bytes
private byte[] foto;
```

Obs. *Esse não é a melhor maneira de hospedar/armazenar imagens, o correto seria hospedar em algum serviço especializado, como por exemplo, **Azure Blob Storage**, **S3** ou "**Buckets**" oferecidos pelo seu cloud provider (AWS, GoogleCloud, Azure entre outros).*

No repositório dessa entidade, teremos que criar métodos de inserção, recuperação e atualização, justamente por conta da anotação "**@JsonIgnore**".

Exemplo:

```
public interface PlantaRepository extends JpaRepository<Planta, Integer> {

    @Modifying
    @Transactional
    @Query("update Planta p set p.foto = ?2 where p.id = ?1")
    void setFoto(Integer id, byte[] foto);

    @Query("select p.foto from Planta p where p.id = ?1")
    byte[] getFoto(Integer id);

    @Modifying
    @Transactional
    @Query("update Planta p set p.relatorioExcel = ?2 where p.id = ?1")
    void setRelatorio(Integer id, byte[] foto);

    @Query("select p.relatorioExcel from Planta p where p.id = ?1")
    byte[] getRelatorio(Integer id);
}
```

Sobre os "endpoints", na prática a construção continua da mesma forma, precisamos informar o que esse endpoints produzem ou consomem, exemplo:

@PostMapping (value = "/foto/{idPlanta}", consumes = "image/*"):

- o parâmetro **consumes** indica que esse endpoint recebe uma imagem de qualquer formato (jpeg, jpg, png e etc).

@GetMapping (value = "/foto/{idPlanta}", produces = MediaType.IMAGE_JPEG_VALUE):

- o parâmetro **produces** indica que esse endpoint retorna ou produz uma imagem com o formato **JPEG**.

@PostMapping (value = "/relatorio/{idPlanta}", consumes = "application/vnd.openxmlformats-officedocument.spreadsheetml.sheet"):

- o parâmetro **consumes** indica o tipo de dado que será aceito no corpo da requisição, o valor passado nele é o que se refere a um mime-type. O mime-type utilizado é arquivo excel com a extensão **xlsx**. uma lista dos tipos de mime-type está em <https://mimetype.io/all-types/> ou em <https://www.sitepoint.com/mime-types-complete-list/>

@GetMapping (value = "/relatorio/{idPlanta}", produces = "application/vnd.openxmlformats-officedocument.spreadsheetml.sheet"):

recupera o relatório (planilha excel, formato **XLSX**) de uma planta.

- **return ResponseEntity.status(200).header("content-disposition", "attachment; filename=\"relatorio-planta.xlsx\").body(relatorio);**
 - esse header **"content-disposition"** indica o nome do arquivo definido padrão em caso de download em navegador e o body será o relatório retornado em **byte**.