

# SUBIDA DE FICHEROS





1.

# MULTIPART CON SPRING

# MULTIPART

- Tipo de mensaje que permite que una petición tenga varias partes delimitadas, con su correspondiente *Content-Type*.

```
MIME-version: 1.0
Content-type: multipart/mixed; boundary="frontera"

This is a multi-part message in MIME format.
--frontera
Content-type: text/plain

Este es el cuerpo del mensaje
--frontera
Content-type: application/octet-stream
Content-transfer-encoding: base64

PGh0bWw+CiAgPGhlYWQ+CiAgPC9oZWFKPgogIDxib2R5PgogICAgPHA+RXN0ZSBlcYBlbCBjdWVy
cG8gZGVsIGlbnNhamU8L3A+CiAgPC9ib2R5Pgo8L2h0bWw+Cic=
--frontera--
```

# MULTIPART

- ▶ De esta forma, podemos enviar los datos del formulario y datos binarios, a la vez.

**<form enctype="multipart/form-data">**

- ▶ El formulario debe incluir un campo de subida de ficheros:

**<input type="file" .../>**

- ▶ Spring soporta perfectamente la gestión de peticiones multiparte.

# MULTIPART CON SPRING

- ▶ Cuando Spring procesa una petición multiparte, nos deja acceder a ella (o ellas) a través de *@RequestParam*

```
@PostMapping("/form")  
public String handleFormUpload(@RequestParam("name") String name,  
                                @RequestParam("file") MultipartFile file) {
```

- ▶ La clase [MultipartFile](#) tiene métodos convenientes para permitirnos procesar el fichero.

# ¿DÓNDE LO **ALMACENAMOS**?

- ▶ Propio proyecto
  - ▶ Fácil para aprender
  - ▶ No es buena práctica en producción
- ▶ Servicio de almacenamiento externo
  - ▶ *GridFS*
  - ▶ Nube (*Amazon, Azure, Drive...*)
  - ▶ Si son imágenes, servicios específicos, como *imgur*.