Spring Boot File Upload

MultipartFile - info

- All we need is MultipartFile object. It contains user send file and all it information.
- Important methods
- getBytes – returns file as a byte array.
- getContentType() – returns a content type (image/png, image/jpeg)
- getName() – name which user send a file
- getOriginalFilename() – origion file name
- getSize() – size of a file in bytes
- getInputStream() – inputStream

Upload file - controller

- All we need to do is get MultipartFile from user and save these file into system.

```
@PostMapping("/upload")
  public ResponseEntity<String> upload( @RequestParam("file") MultipartFile file) {
      String fileName = attachService.saveToSystem(file);
      return ResponseEntity.ok().buody(fileName);
  }
```

Upload file - service 1

- Save file with the origin name
- Simple way to save.

```
public String  saveToSystem(MultipartFile file) {
        try {
            File folder = new File("attaches");
            if (!folder.exists()) {
                folder.mkdir();
            }

            byte[] bytes = file.getBytes();
            Path path = Paths.get("attaches/" + file.getOriginalFilename());
            Files.write(path, bytes);
            return file.getOriginalFilename();
        } catch (IOException e) {
            e.printStackTrace();
        }
        return null;
    }
```

Upload file - service 2

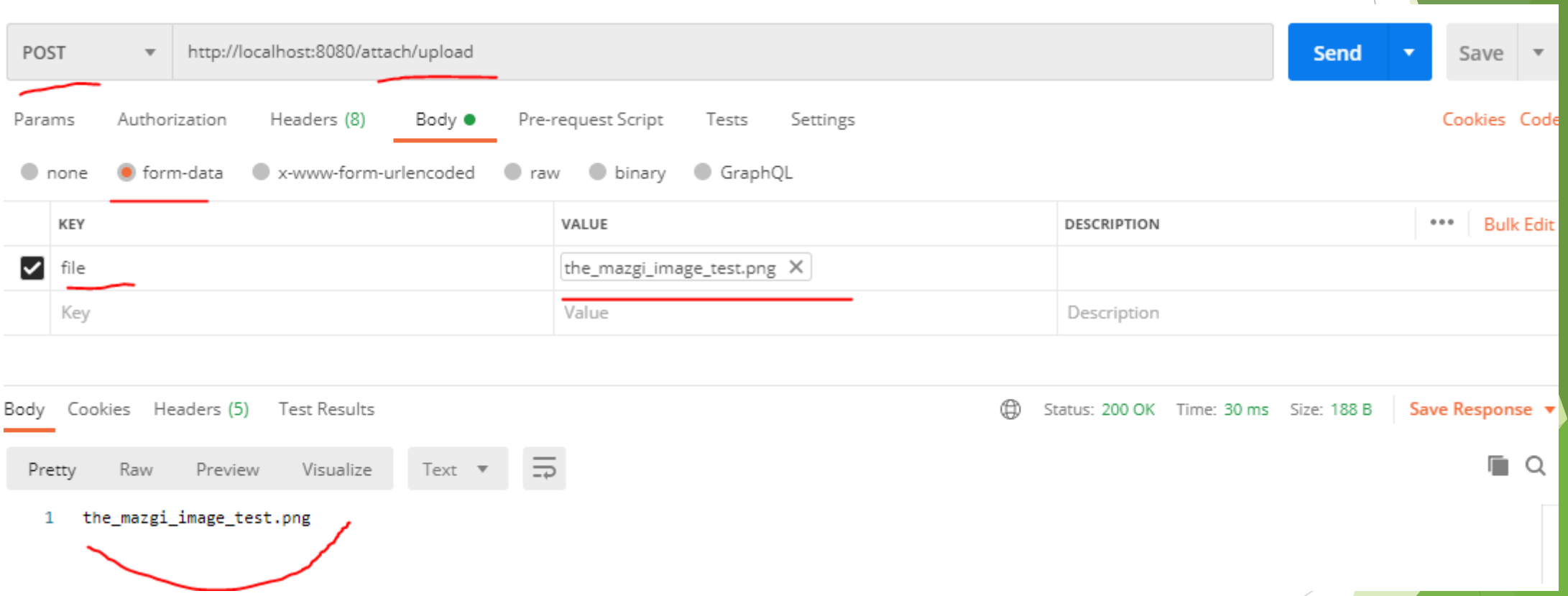
- Save file with the origin name
- Another way of saving file to system

- Mazgi bu 2chi boshqacha varianti

```
public String saveToSystem(MultipartFile file) {
    try {
        Path copyLocation = Paths.get("attaches" + File.separator + file.getOriginalFilename());
        Files.copy(file.getInputStream(), copyLocation, StandardCopyOption.REPLACE_EXISTING);
        return file.getOriginalFilename();
    } catch (IOException e) {
        e.printStackTrace();
    }
    return null;
}
```

MultipartFile – send using PostMan

- How to send file upload using postman



dasturlash.uz

MultipartFile – send using .http

- In .http file

```
POST http://localhost:8080/attach/upload
Content-Type: multipart/form-data; boundary=WebAppBoundary

--WebAppBoundary
Content-Disposition: form-data; name="file"; filename="IMG_3625.MP4"

< D:\mazgifolder\IMG_3625.MP4
```

- Mazgi choose write location and name of the file

Load/Download image/attach

- The ide bunaqa frontend file unique nimadirini serverga berib yuboradi va server shu file ni olib uni byte ko’rinishida frontend ga return qiladi.
- Bizni xolatda hozir oddiy image name dan foydalanamiz.

Load/Download – controller as byte[]

- In this case file will be opened in browser

```
@GetMapping(value = "/open/{fileName:.+}", produces = MediaType.IMAGE_PNG_VALUE)
    public byte[] open(@PathVariable("fileName") String fileName) {
        if (fileName != null && fileName.length() > 0) {
            try {
                return this.attachService.loadImage(fileName);
            } catch (Exception e) {
                e.printStackTrace();
                return new byte[0];
            }
        }
        return null;
    }
```

Load/Download – service as byte[]

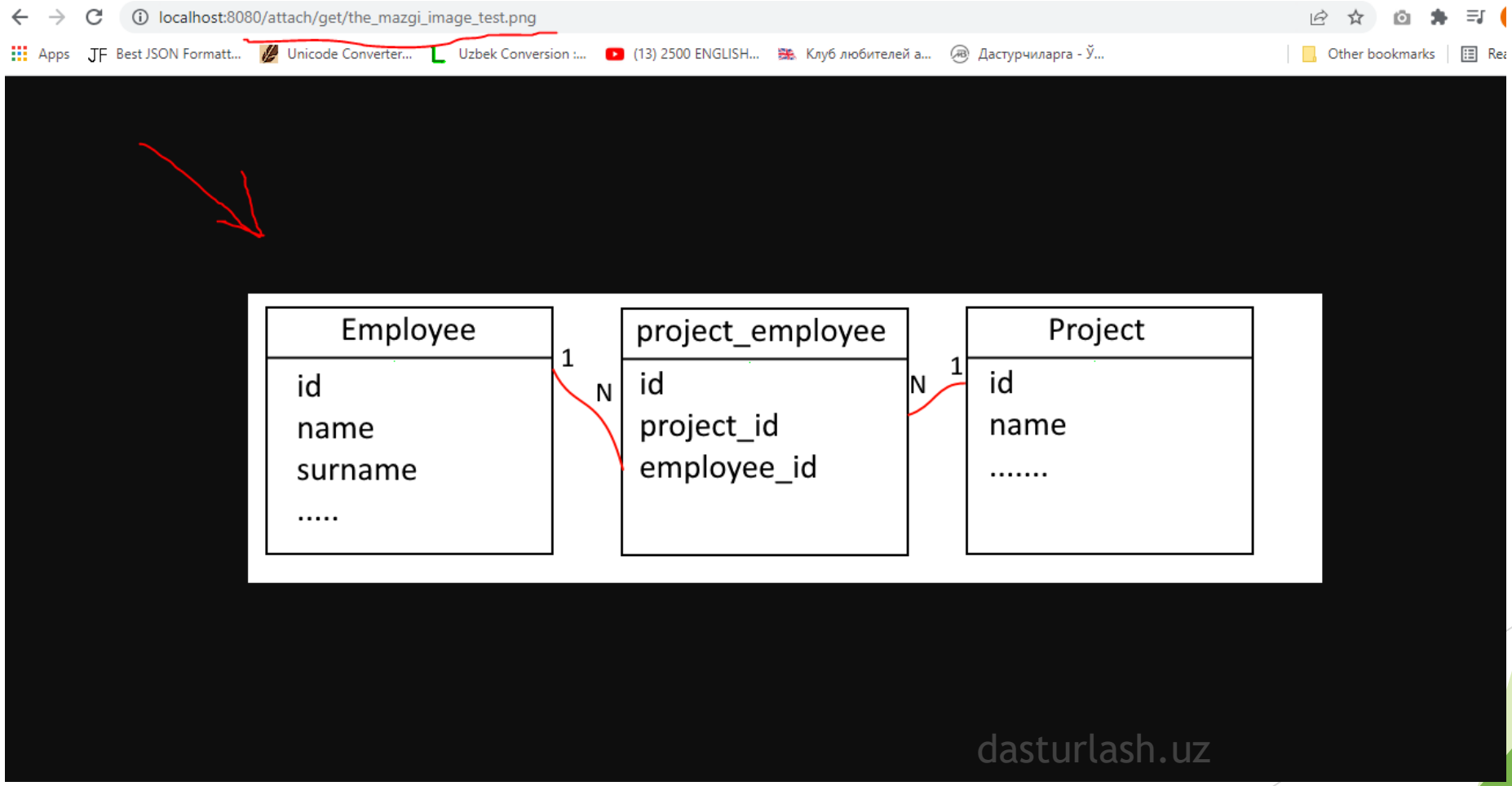
```
public byte[] loadImage(String fileName) {
        byte[] imageInByte;

        BufferedImage originalImage;
        try {
            originalImage = ImageIO.read(new File("attaches/" + fileName));
        } catch (Exception e) {
            return new byte[0];
        }

        ByteArrayOutputStream baos = new ByteArrayOutputStream();
        ImageIO.write(originalImage, "png", baos);

        baos.flush();
        imageInByte = baos.toByteArray();
        baos.close();
        return imageInByte;
}
```

Load/Download – front example



dasturlash.uz

Load/Download – Html example

- Put correct url and open it as html file.

```
<img src="http://localhost:8080/attach/open/test.jpg"></img>
```

Load/Download any kind of file

Load/Download – Any file as byte[] - Controller

- Bu Controller dagi method barcha turdagi file,rasm,video,… larni byte ko’rinishda return qiladi.

```
@GetMapping(value = "/open_general/{fileName}", produces = MediaType.ALL_VALUE)
   public byte[] open_general(@PathVariable("fileName") String fileName) {
       return attachService.open_general(fileName);
   }
```

Load/Download – Any file as byte[] –Service 1

- Bu service dagi method barcha turdagi file ni sistemedan o’qib uni file ko’rinishida return qiladi.

```
public byte[] open_general(String key) {
        byte[] data;
        try {
            AttachEntity entity = get(key);
            String path = entity.getPath() + "/" + key + "." + entity.getExtension();
            Path file = Paths.get("uploads/" + path);
            data = Files.readAllBytes(file);
            return data;
        } catch (IOException e) {
            e.printStackTrace();
        }
        return new byte[0];
    }
```

Load/Download – Any file as byte[] –Service 2

- Bu sistemadan file ni byte ko’rinishida o’qishni 2chi usuli.

```
public byte[] open_general2(String fileName) {
        File file = new File("uploads/" + fileName);
        FileInputStream fileInputStream = null;
        byte[] bytes = new byte[(int) file.length()];
        try {
            fileInputStream = new FileInputStream(file);
            fileInputStream.read(bytes);
            fileInputStream.close();
            return bytes;
        } catch (Exception e) {
            e.printStackTrace();
        }
        return new byte[0];
    }
```

- Sistemedan file ni byte[] ko’rinishida o’qishni yo’llari juda ko’p Mazgi.

Load/Download – open in html

- Mazgi change url with yours. Do not be DML

- Image

```
<img src="http://localhost:8080/attach/open_general/test.jpg"></img>
```

- Video

```
<video width="320" height="240" controls>
    <source src="http://localhost:8080/attach/open_general/IMG_4593.MP4" type="video/mp4">
    Your browser does not support the video tag.
</video>
```

- Audio/mp3

```
<audio controls autoplay>
    <source src="http://localhost:8080/attach/open_general/857473710640.m4a" type="audio/ogg">
    Your browser does not support the audio element.
</audio>
```

Downloadable File

Load/Download – controller as downloadable file

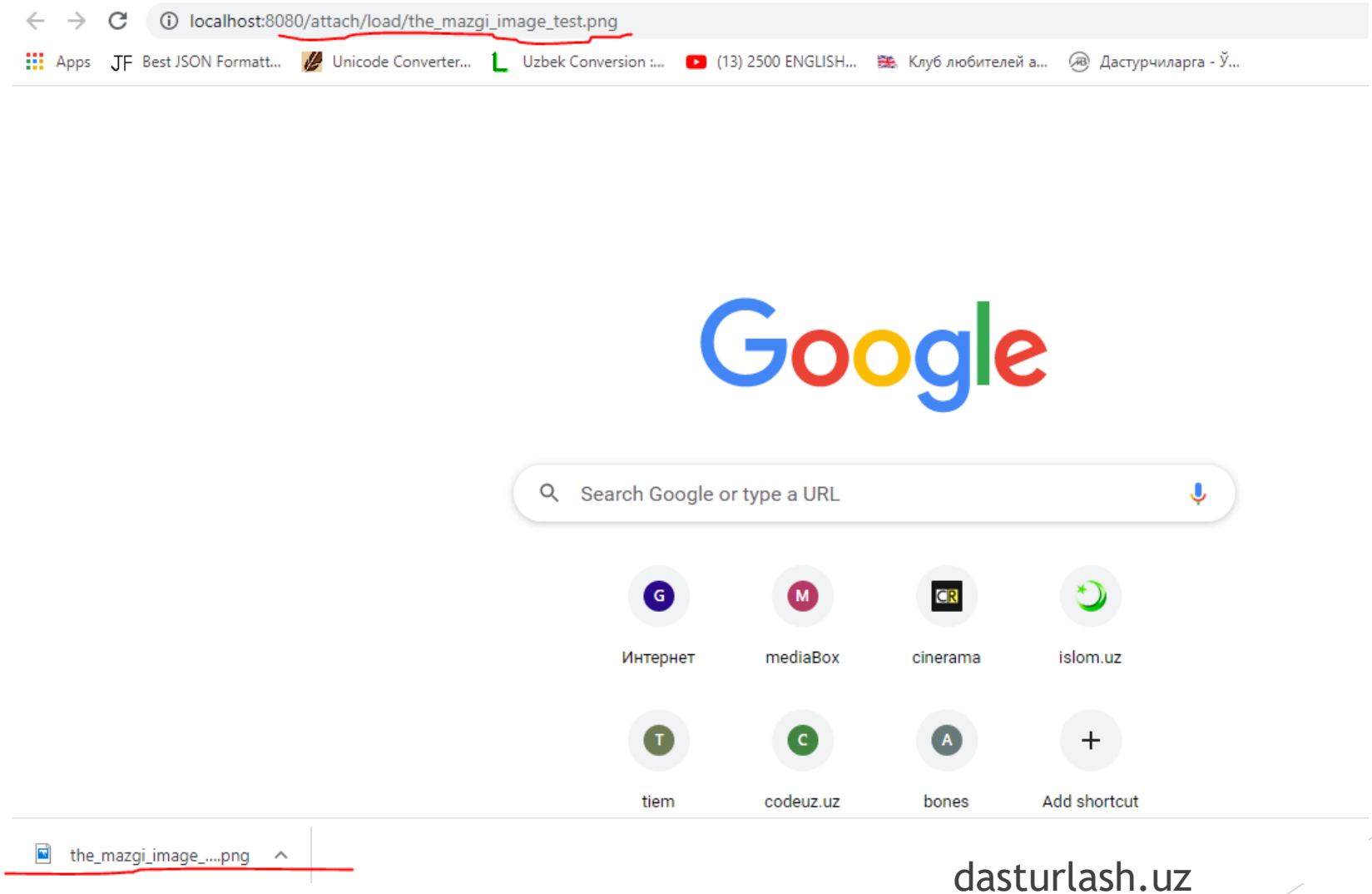
```
@GetMapping("/download/{fineName:.+}")
  public ResponseEntity<Resource> download(@PathVariable("fineName") String fileName) {
      Resource file = attachService.download(fileName);
      return ResponseEntity.ok().header(HttpHeaders.CONTENT_DISPOSITION,
            "attachment; filename=\"" + file.getFilename() + "\"").body(file);
  }
```

Load/Download – service as downloadable file

```
public Resource download(String fileName) {
        try {
            Path file = Paths.get("attaches/" + fileName);
            Resource resource = new UrlResource(file.toUri());

            if (resource.exists() || resource.isReadable()) {
                return resource;
            } else {
                throw new RuntimeException("Could not read the file!");
            }
        } catch (MalformedURLException e) {
            throw new RuntimeException("Error: " + e.getMessage());
        }
    }
```

Load/Download – frontend as downloadable example



dasturlash.uz

Load/Download – html as downloadable example

- In html file.

```
<a href="http://localhost:8080/attach/download/test.jpg">Download</a>
```

Mazgi note

- In above example we can download any kind of file.

Important

- Namunadagiday qilib hamam rasm, file larni bitta papkaga save qilish yaxshi emas. 1 ta papkada 10,000 da file bo’lishi mumkin. Bu sistemaga qarab.
- Zo’r bo’lad agar sana bo’yicha save qilsak.
- Buni barchasi spring_boot_attach_example – projectida.

Upload file - In date system 1

- Create method for generating

```
public String getYmDString() {
        int year = Calendar.getInstance().get(Calendar.YEAR);
        int month = Calendar.getInstance().get(Calendar.MONTH) + 1;
        int day = Calendar.getInstance().get(Calendar.DATE);

        return year + "/" + month + "/" + day; // 2022/04/23
    }
```

Upload file - In date system 2

- Create method for getting extension of a file.

```
public String getExtension(String fileName) { // mp3/jpg/npg/mp4.....
        int lastIndex = fileName.lastIndexOf(".");
        return fileName.substring(lastIndex + 1);
    }
```

Upload file - In date system 3

- Inside upload method 1.
- Prepare file for saving in different folder and with different name.

```
String pathFolder = getYmDString(); // 2022/04/23
      File folder = new File("uploads/" + pathFolder);
      if (!folder.exists()) {
          folder.mkdirs();
      }

 String key = UUID.randomUUID().toString(); // dasdasd-dasdasda-asdasda-asdasd
 String extension = getExtension(file.getOriginalFilename()); // dasda.asdas.dasd.jpg
```

Upload file - In date system 3

- Inside upload method 2.
- Save file into system and save AttachEntity.

```
    // save attachEntity
   // prepare dto for return.
try{
        byte[] bytes = file.getBytes();
        Path path = Paths.get("uploads/" + pathFolder + "/" + key + "." + extension);
        Files.write(path, bytes);
   } catch (IOException e) {
        e.printStackTrace();
   }
// return anything you want mazgi.
```

Multipart File Size

- By default, Spring Boot max file upload size is 1MB, you can configure the values via following application properties :
- spring.http.multipart.max-file-size=10MB
- spring.http.multipart.max-request-size=10MB
- In Spring Boot user the following:

spring.servlet.multipart.max-file-size=10MB
spring.servlet.multipart.max-request-size=10MB

Links

- https://www.bezkoder.com/spring-boot-file-upload/
- https://mkyong.com/spring-boot/spring-boot-file-upload-example/
- https://stackabuse.com/uploading-files-with-spring-boot/
- https://www.techgeeknext.com/spring-boot/spring-boot-upload-file