

[Spring] PostgreSQL

bytea

()

snippet.sql

```
CREATE TABLE public.attach_file_info
(
    file_size INTEGER NOT NULL, --
    file_type CHARACTER VARYING(4), --
    thumbnail text, --
    file bytea,
)
```

```
file      bytea
```

snippet.java

```
private void fileUploadWithThumbnail(String attach_file_seq, String
file_path, String user_id, String thumbnail) {
    File uploadFile = new File(file_path);
    MultipartEntityBuilder builder = MultipartEntityBuilder.create() //
        .setCharset(Charset.forName("UTF-8")) // UTF-8
        .setMode(HttpMultipartMode.BROWSER_COMPATIBLE);
    builder.addPart("file", new FileBody(uploadFile)); // FileBody
        File
    builder.addTextBody("attach_file_seq", attach_file_seq,
    ContentType.create("Multipart/related", "UTF-8"));
    builder.addTextBody("user_id", user_id,
    ContentType.create("Multipart/related", "UTF-8"));
    builder.addTextBody("thumbnail", thumbnail,
    ContentType.create("Multipart/related", "UTF-8"));

    try {
        InputStream inputStream = null;
        HttpClient httpClient = new DefaultHttpClient();
        HttpPost post = new HttpPost(server_domain +
        "/file/fileUploadDisk.json"); // URL
        HttpEntity multipart = builder.build();
```

```
        post.setEntity(multipart);
        HttpResponse httpResponse = httpClient.execute(post);
        HttpEntity httpEntity = httpResponse.getEntity();
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

MultipartEntityBuilder

- file
- 가
- Post

snippet.java

```
@RequestMapping("/fileUploadDisk.json")
public void fileUploadDisk(RMap rmap, ModelMap model,
@RequestParam("file") MultipartFile file) throws
IOException {
    // 가 가
    if (file.isEmpty())
        return;

    rmap.put("file", file.getBytes());
    commonFileService.insertAttachFileInfo(rmap, model);
}
```

getBytes byte array mybatis

insert Query

snippet.sql

```
<!-- -->
<INSERT id="insertAttachFileInfo">
    /* file.insertAttachFileInfo */
    INSERT INTO ATTACH_FILE_INFO
    (
        thumbnail,
        file
    )
    VALUES
```

```
(  
    #{thumbnail},  
    #{file}  
)  
</INSERT>
```

byte array

snippet.javascript

```
@RequestMapping("/attachImage.do")  
public void attachImage(RMap rmap, ModelMap model, HttpServletRequest  
request, HttpServletResponse response) throws IOException {  
    //  
    UMap umap = commonService.selectAttachFile(rmap, model);  
  
    byte[] data = (byte[]) umap.get("file");  
    String file_type = (String)umap.get("file_type");  
  
    if (data == null || !file_type.equals("0001")){  
        String thumbnail = (String) umap.get("thumbnail");  
        data = Base64.decodeBase64(thumbnail.split("base64,")[1]);  
    }  
  
    //  
    response.setContentType("image/jpeg");  
    response.getOutputStream().write(data);  
}
```

snippet.java

```
@RequestMapping("/fileDownload.do")  
public void fileDownload(RMap rmap, ModelMap model, HttpServletRequest  
request, HttpServletResponse response) throws IOException {  
    UMap umap = commonService.selectAttachFile(rmap, model);  
  
    String filename = (String) umap.get("original_file_name");  
    String fileNm = new String(filename.getBytes("UTF-8"),  
"ISO-8859-1");  
  
    response.setContentType("application/octet-stream");
```

```
response.setHeader("Content-Disposition", "attachment;filename=" +  
fileNm);  
  
byte[] data = (byte[]) umap.get("file");  
response.getOutputStream().write(data);  
}
```

Plugin Backlinks:

From:
<https://jace.link/> - Various Ways

Permanent link:
<https://jace.link/open/spring-postgresql-%EC%97%90-%ED%8C%8C%EC%9D%BC-%EC%A0%80%EC%9E%A5%ED%95%98%EA%B3%A0-%EB%B6%88%EB%9F%AC%EC%98%A4%EA%B8%B0>

Last update: 2020/06/02 09:25

