

主題 6：檔案上傳及下載

提供一個 RestController

- 名稱為【DemoController】，其功能包括
 1. 檔案上傳 uploadAttachmentInDemo
 2. 檔案下載 downloadAttachmentInDemo
- 上傳後的檔案會儲存NAS空間 【 D:\workspace-IntelliJ\nfsdata 】
- 將檔名及相關資料記錄在一張資料表 【 ATTACHMENT(上傳附件清單) 】

DownloadAttachmentDto.java

```
package com.fubonlife.mytest.common.model.demo;

import com.google.common.base.Charsets;
import lombok.Builder;
import lombok.Data;
import lombok.extern.slf4j.Slf4j;
import org.apache.commons.lang3.StringUtils;
import org.springframework.core.io.Resource;
import org.springframework.http.MediaType;

import java.io.UnsupportedEncodingException;
import java.net.URLEncoder;

@Data
@Builder
@Slf4j
public class DownloadAttachmentDto {
    private String fileName;
    private MediaType contentType;
    private Resource fileResource;

    //
    public String convertContentDisposition() {
        String contentDisposition = "";
        try {
            contentDisposition = URLEncoder.encode(
                this.getFileName(), Charsets.UTF_8.name());
            contentDisposition = StringUtils.
                replace(contentDisposition, "+", "%20");
        } catch (UnsupportedEncodingException e) {
            log.error("DownloadAttachmentDto url
                encode , {}", this.getFileName());
        }
        return contentDisposition;
    }
}
```

DemoController.java

```
package com.fubonlife.mytest.api.controller;

import com.fubonlife.mytest.common.model.demo.
DownloadAttachmentDto;
import com.fubonlife.mytest.common.service.
DemoService;
import io.swagger.annotations.Api;
import io.swagger.annotations.ApiOperation;
import io.swagger.annotations.ApiParam;
import org.springframework.beans.factory.
annotation.Autowired;
import org.springframework.core.io.Resource;
import org.springframework.http.ResponseEntity;
import org.springframework.util.ObjectUtils;
import org.springframework.web.bind.annotation.
PostMapping;
import org.springframework.web.bind.annotation.
RequestMapping;
import org.springframework.web.bind.annotation.
RequestParam;
import org.springframework.web.bind.annotation.
RestController;
import org.springframework.web.multipart.
MultipartFile;

@RestController
@RequestMapping(value = "/api/demo")
@Api(tags = {"Demo"}, description = "/")
public class DemoController {

    @Autowired
    DemoService demoService;

    @ApiOperation("DME0")
    @PostMapping("/attachment/upload")
    public String uploadAttachmentInDemo
(@RequestParam @ApiParam(value="", required =
true) MultipartFile file) {

        return demoService.uploadAttachment(file);
    }

    @ApiOperation("DME0")
    @PostMapping("/attachment/download")
    public ResponseEntity<Resource>
downloadAttachmentInDemo(@RequestParam @ApiParam
(value="", required = true) String attachmentId) {

        DownloadAttachmentDto result = demoService.
downloadAttachment(attachmentId);

        return ObjectUtils.isEmpty(result) ?
ResponseEntity.notFound().build() : ResponseEntity.
ok()
            .contentType(result.getContentType())
            .header("Access-Control-Expose-
Headers", "Content-Disposition")
            .header("Content-Disposition", String.
format("attachment; filename=\"%s\"", result.
convertContentDisposition()))
            .body(result.getFileResource());
    }
}
```

DemoServiceImpl.java

```
package com.fubonlife.mytest.common.service.impl;

import com.fubonlife.mytest.common.entity.primary.Attachment;
import com.fubonlife.mytest.common.model.demo.DownloadAttachmentDto;
import com.fubonlife.mytest.common.repository.primary.AttachmentRepository;
import com.fubonlife.mytest.common.service.AttachmentService;
import com.fubonlife.mytest.common.service.DemoService;
import com.fubonlife.mytest.common.service.NasService;
import lombok.extern.slf4j.Slf4j;
import org.apache.commons.lang3.StringUtils;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.http.MediaType;
import org.springframework.stereotype.Service;
import org.springframework.web.multipart.MultipartFile;

import java.io.IOException;
import java.nio.file.Files;
import java.nio.file.Path;
import java.nio.file.Paths;

@Service
@Slf4j
public class DemoServiceImpl implements DemoService {

    @Autowired
    NasService nasService;

    @Autowired
    AttachmentService attachmentService;

    @Autowired
    AttachmentRepository attachmentRepository;

    @Override
    public String uploadAttachment(MultipartFile file) {
        // NAS
        String filePath = nasService.uploadFile(file);

        // +NASATTACHMENT()
        if(StringUtils.isNotBlank(filePath)) {
            Attachment attachment = attachmentService.Save(file, filePath);
            return attachment.getAttachmentId();
        }
        return null;
    }

    @Override
    public DownloadAttachmentDto downloadAttachment(String attachmentId) {
        // ATTACHMENT()
        Attachment attachment = attachmentRepository.findById(attachmentId).orElse(null);
        if(attachment != null) {
            return DownloadAttachmentDto.builder()
                .fileResource(nasService.downloadFile(attachment.getFilePath())) // NAS
                .fileName(attachment.getFileName()) //
                .contentType(getContentType(attachment)) //
                .build();
        }
        return null;
    }

    private MediaType getContentType(Attachment attachment) {
        Path path = Paths.get(attachment.getFilePath());
        MediaType contentType = null;
        try {
            contentType = MediaType.valueOf(Files.probeContentType(path));
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

```

    }
    return contentType;
}
}

```

NasServiceImpl.java

```

package com.fubonlife.mytest.common.service.impl;

import com.fubonlife.mytest.common.service.NasService;
import com.fubonlife.mytest.common.util.FileUtil;
import lombok.extern.slf4j.Slf4j;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.beans.factory.annotation.Value;
import org.springframework.core.io.InputStreamResource;
import org.springframework.core.io.Resource;
import org.springframework.stereotype.Service;
import org.springframework.web.multipart.MultipartFile;

import java.io.FileInputStream;
import java.nio.file.Files;
import java.nio.file.Path;
import java.nio.file.Paths;
import java.nio.file.StandardCopyOption;
import java.util.UUID;

@Service
@Slf4j
public class NasServiceImpl implements NasService {

    @Value("${nas.upload-path}")
    String nasPathStr;

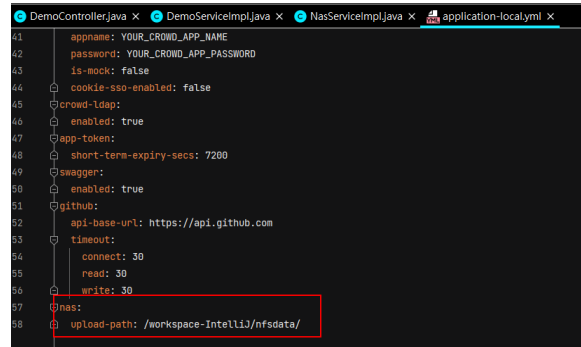
    // NAS
    @PostConstruct
    private void createNASDirectories() throws IOException {
        Path nasPath = Paths.get(nasPathStr);

        // NAS(OS=WIND:\workspace-IntelliJ\nfsdata)
        if (!Files.exists(nasPath)) {
            Files.createDirectories(nasPath);
        }
    }

    //
    @Override
    public String uploadFile(MultipartFile file) {
        Path nasPath = Paths.get(nasPathStr);
        try {
            // NASUUID(ATTACHMENT)
            nasPath = nasPath.resolve(UUID.randomUUID() + "." + FileUtil.getExtension(file));

            // NAS
            Files.copy(file.getInputStream(), nasPath, StandardCopyOption.REPLACE_EXISTING);
        } catch (Exception e) {
            log.error("upload {} to NAS fail", file.getOriginalFilename(), e);
            return null;
        }
    }
}

```



```

61  appname: YOUR_CROWD_APP_NAME
62  password: YOUR_CROWD_APP_PASSWORD
63  is-mock: false
64  cookie-sso-enabled: false
65  crowd-ldap:
66  enabled: true
67  app-token:
68  short-term-expiry-secs: 7200
69  swagger:
70  enabled: true
71  github:
72  api-base-url: https://api.github.com
73  timeout:
74  connect: 30
75  read: 30
76  write: 30
77  @nas:
78  upload-path: /workspace-IntelliJ/nfsdata/

```

```
        }
        return nasPath.toString();
    }

    //
    @Override
    public Resource downloadFile(String filePath) {
        FileInputStream fileInputStream = null;
        try {
            fileInputStream = new FileInputStream
(filePath);
        } catch (Exception e) {
            log.error("download {} from NAS fail",
filePath, e);
        }
        return fileInputStream == null ? null :
new InputStreamResource(fileInputStream);
    }
}
```