

数字底座-文件组件使用文档

产品版本：V1.0

文档版本：202401

北京有生博大软件股份有限公司

(1) 添加数字底座 maven 仓库地址

```

<repositories>
  <repository>
    <id>nexus</id>
    <name>local private nexus</name>
    <url>https://svn.youshengyun.com:9900/nexus/repository/maven-
public/</url>
    <snapshots>
      <updatePolicy>always</updatePolicy>
    </snapshots>
  </repository>
</repositories>
<pluginRepositories>
  <pluginRepository>
    <id>nexus</id>
    <name>local private nexus</name>
    <url>https://svn.youshengyun.com:9900/nexus/repository/maven-
public/</url>
  </pluginRepository>
</pluginRepositories>

```

(2) 引入依赖包

```
<!-- 文件组件-开始 -->
```

```

<dependency>
  <groupId>net.risesoft</groupId>
  <artifactId>risenet-y9boot-support-fileService-FTP</artifactId>
  <version>v9.6.3-SNAPSHOT</version>
</dependency>

```

```
<!-- 文件组件-结束 -->
```

包说明：这里没有引入对接数字底座单点登录的包，所以记录文件上传信息至数字底座表中时，没有记录上传人信息

(3) 添加配置信息

y9:

#系统名称，用于记录文件是哪个系统上传的

systemName: demo-file

feature:

jpa:

#数字底座文件上传的包的扫描的包路径，不能更改

packagesToScanEntityPublic: net.risesoft.y9public.entity

packagesToScanRepositoryPublic:

net.risesoft.y9public.repository

file:

base64FileName: false

encryptionFileContent: false

#privateKey:

#publicKey:

#ftp 配置信息

ftp:

host: localhost

port: 21

username: y9admin

password: '111111'

blockWhenExhausted: true

```

        bufferSize: 10240
        connectTimeout: 50000
        controlEncoding: UTF-8
        dataTimeout: 1200000
        fileType: 2
        maxIdle: 10
        maxTotal: 50
        minIdle: 2
        maxWaitMillis: 5400000
        testOnBorrow: true
        testOnCreate: true
        testOnReturn: true
        testWhileIdle: true
        useEPSVwithIPv4: false
        passiveMode: true
spring:
  #文件相关表信息需要的数据源信息，需要配置为数字底座的公共数据源
  datasource:
    druid:
      y9-public:
        driver-class-name: com.mysql.cj.jdbc.Driver
        url:
jdbc:mysql://localhost:3306/y9_public?serverTimezone=GMT%2B8&nullCatalogMea
nsCurrent=true&useUnicode=true&characterEncoding=utf-
8&rewriteBatchedStatements=true&useCompression=true&useSSL=false&allowPubli
cKeyRetrieval=true
        username: root
        password: 111111
        initialSize: 1
        maxActive: 20
        maxPoolPreparedStatementPerConnectionSize: 100
        maxWait: 60000
        minEvictableIdleTimeMillis: 300000
        minIdle: 1
        poolPreparedStatements: true
        testOnBorrow: false
        testOnReturn: false
        testWhileIdle: true
        timeBetweenEvictionRunsMillis: 60000
        useGlobalDataSourceStat: true
        validationQuery: SELECT 1 FROM DUAL
  #redis 配置，主要用于雪花算法生成 id
  redis:
    database: 8
    host: localhost
    lettuce:
      pool:
        max-active: 8
        max-idle: 8
        max-wait: -1
        min-idle: 0
      shutdown-timeout: 100ms
    password: y9i-83204585
    port: 6379
    ssl: false
    timeout: 10000

```

(4) 代码示例

文件对接示例工程码云地址：<https://gitee.com/risesoft-y9/y9-core.git>
[/example/risenet-y9demo-file](#)

注意：该示例工程建立在数字底座已运行的基础上。

```
package net.risesoft.controller;

import java.io.OutputStream;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import org.apache.commons.lang3.StringUtils;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import org.springframework.beans.factory.annotation.Autowired;
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;

import net.risesoft.y9public.entity.Y9FileStore;
import net.risesoft.y9public.service.Y9FileStoreService;

@RestController
@RequestMapping(value = "/main")
public class MainController {

    Logger log = LoggerFactory.getLogger(MainController.class);

    @Autowired
    private Y9FileStoreService y9FileStoreService;

    /**
     * 上传文件,业务系统需要存储文件系统返回的文件唯一标示,该文件唯一标示可以用来下载和删除对应的文件
     *
     * @param file
     */
    @PostMapping(value = "/uploadFile")
    public void uploadFile(MultipartFile file) {
        try {
            String fullPath = "/aaaa/bbbb";
            Y9FileStore y9FileStore = y9FileStoreService.uploadFile(file, fullPath,
file.getOriginalFilename());
            log.info("fileStoreId:{", y9FileStore.getId());
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    /**
     * 根据文件唯一标示下载文件
     *
     * @param fileStoreId
     * @param response
     * @param request
     */
    @RequestMapping(value = "/download")
    public void download(@RequestParam String fileStoreId, HttpServletResponse response,
HttpServletRequest request) {
        try {
            Y9FileStore y9FileStore = y9FileStoreService.getById(fileStoreId);
            String title = y9FileStore.getFileName();
            title = java.net.URLEncoder.encode(title, "UTF-8");
            title = StringUtils.replace(title, "+", "%20");// 替换空格
            response.reset();
            response.setHeader("Content-disposition", "attachment; filename=\"" + title +
"\");

            response.setHeader("Content-type", "text/html;charset=UTF-8");
            response.setContentType("application/octet-stream");
            OutputStream out = response.getOutputStream();
```

```
        y9FileStoreService.downloadFileToOutputStream(fileStoreId, out);
        out.flush();
        out.close();
    } catch (Exception e) {
        e.printStackTrace();
    }
}

/**
 * 根据文件唯一标示删除文件
 *
 * @param fileStoreId
 */
@RequestMapping(value = "/deleteFile")
public void deleteFile(@RequestParam String fileStoreId) {
    try {
        y9FileStoreService.deleteFile(fileStoreId);
    } catch (Exception e) {
        e.printStackTrace();
    }
}
}
```