

1. 使用

话不多说，直接进入主题~

目前公司内部文件预览系统已经部署到正式环境，并应用到线上项目中。接入只需要简单的两个步骤如果你有文件在线预览的需求非常建议你使用~

这里就以远智学堂-资料下载的文件预览功能为例，介绍项目中如何接入使用文件预览系统：

更详细的使用请参考官方文档：[使用指南](#)、[接入说明](#)。

1.1. 前端使用

```
import { Base64 } from 'js-base64';

window.open('preview/onlinePreview?url=' + encodeURIComponent(Base64.encode(url)));
```

url 参数为文件资源线上地址，通过 **Base64** 进行编码传递，使用**相对地址**进行跳转。

1.2. nginx 配置

预览系统部署在公司内网，外网无法访问，所以外网访问需要配置代理。

```
location ^~ /school-paper/preview/ {
    proxy_pass http://172.18.178.141:8012/preview/;
    proxy_set_header Host $host;
    proxy_set_header X-Real-IP $remote_addr;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    proxy_set_header X-Forwarded-Proto $scheme;
}
```

在项目对应的 nginx 服务配置中，将 `/preview/`（可根据项目实际进行修改）代理转发到文件预览系统，目前文件预览系统部署在公司内网 `172.18.178.141:8012`。

过程示例：

假如 url 为：<http://img2.yzwill.cn/course/157346212394145626/C9216CD502AB485FB1B99F8BC8125406.pdf>

项目页面访问地址为：<http://pre-bst.yzwill.cn/school-paper/index.html#/classDataDownload/index>

文件预览时跳转的访问地址为：[http://pre-bst.yzwill.cn/school-paper/preview/onlinePreview?](http://pre-bst.yzwill.cn/school-paper/preview/onlinePreview?url=aHR0cDovL2ltZzluaXp3aWxsLmNuL2NvdXJzZS8xNTczNDYyMTIzOTQxNDU2MjYvNTQyNzA2MUFCQTZFNEY0QkI5NzZDQzZGMDg0REIxMzAucGRm)

[url=aHR0cDovL2ltZzluaXp3aWxsLmNuL2NvdXJzZS8xNTczNDYyMTIzOTQxNDU2MjYvNTQyNzA2MUFCQTZFNEY0QkI5NzZDQzZGMDg0REIxMzAucGRm](http://pre-bst.yzwill.cn/school-paper/preview/onlinePreview?url=aHR0cDovL2ltZzluaXp3aWxsLmNuL2NvdXJzZS8xNTczNDYyMTIzOTQxNDU2MjYvNTQyNzA2MUFCQTZFNEY0QkI5NzZDQzZGMDg0REIxMzAucGRm)

实际访问地址为：[http://172.18.178.141:8012/preview/onlinePreview?](http://172.18.178.141:8012/preview/onlinePreview?url=aHR0cDovL2ltZzluaXp3aWxsLmNuL2NvdXJzZS8xNTczNDYyMTIzOTQxNDU2MjYvNTQyNzA2MUFCQTZFNEY0QkI5NzZDQzZGMDg0REIxMzAucGRm)

[url=aHR0cDovL2ltZzluaXp3aWxsLmNuL2NvdXJzZS8xNTczNDYyMTIzOTQxNDU2MjYvNTQyNzA2MUFCQTZFNEY0QkI5NzZDQzZGMDg0REIxMzAucGRm](http://172.18.178.141:8012/preview/onlinePreview?url=aHR0cDovL2ltZzluaXp3aWxsLmNuL2NvdXJzZS8xNTczNDYyMTIzOTQxNDU2MjYvNTQyNzA2MUFCQTZFNEY0QkI5NzZDQzZGMDg0REIxMzAucGRm)

2. 背景

在一次项目需求中，需要实现一个文件在线预览的功能，根据接口返回的文件资源线上资源地址，需要将文件展示在页面上进行预览。查看了以往项目中的实现方式大体有如下方式：

2.1. 前端处理

目前大部分项目中采用的是通过判断文件的类型，然后根据不同类型的文件加载不同的第三方依赖进行解析展示，如 `docx-preview`、`vue-office` 等等，而在多个不同项目中实现功能都需要同样的进行引用依赖使用，这样不仅增加了代码量，而且也不利于后期维护。

2.2. 免费的第三方在线预览服务

有些项目直接使用了第三方的在线预览服务，如微软 [https://view.officeapps.live.com/op/view.aspx?](https://view.officeapps.live.com/op/view.aspx?src=${encodeURIComponent(url)})

`src=${encodeURIComponent(url)}`，支持常用的 `docx`、`pptx`、`xlsx` 文件类型，使用也比较方便，直接传入文件地址即可，但缺点是加载比较慢，偶尔会超时崩溃，访问不稳定，导致预览效果并不理想，而且确保不了文件的保密性。

2.3. 付费的第三方在线预览服务

在现有项目中使用的付费第三方在线预览服务是阿里云 IMM，支持多种文件类型，应用覆盖全终端，安全性高，但缺点是需要付费，还需要对接服务。

根据目前这三种的实现方式，沿用到新项目都不太理想，所以需求很明显了。

期望：

- 通用性高、简单易用
- 支持更多的文件类型预览
- 稳定可控
- 文件的保密性
- 成本低

于是有了基于 **kkFileView** 开源项目来部署了一套公司内部的通用的文件在线预览系统。

3. kkFileView 介绍

kkFileView 文档在线预览项目解决方案，项目使用流行的 **spring boot** 搭建，易上手和部署。万能的文件预览开源项目，基本支持主流文档格式预览支持的文件预览格式非常丰富：

1. 支持 doc, docx, xls, xlsx, xlsx, ppt, pptx, csv, tsv, dotm, xlt, xltm, dot, dotx, xlam, xla 等 Office 办公文档
2. 支持 wps, dps, et, ett, wpt 等国产 WPS Office 办公文档
3. 支持 odt, ods, ots, odp, otp, six, ott, fodt, fods 等OpenOffice、LibreOffice 办公文档
4. 支持 vsd, vsd, vss, vst, vstx 等 Visio 流程图文件
5. 支持 wmf, emf 等 Windows 系统图像文件
6. 支持 psd 等 Photoshop 软件模型文件
7. 支持 pdf, ofd, rtf 等文档
8. 支持 xmind 软件模型文件
9. 支持 bpmn 工作流文件
10. 支持 eml 邮件文件
11. 支持 epub 图书文档
12. 支持 obj, 3ds, stl, ply, gltf, glb, off, 3dm, fbx, dae, wrl, 3mf, ifc, brep, step, iges, fcstd, bim 等 3D 模型文件
13. 支持 dwg, dxf 等 CAD 模型文件
14. 支持 txt, xml(渲染), md(渲染), java, php, py, js, css 等所有纯文本
15. 支持 zip, rar, jar, tar, gzip, 7z 等压缩包
16. 支持 jpg, jpeg, png, gif, bmp, ico, jfif, webp 等图片预览（翻转，缩放，镜像）
17. 支持 tif, tiff 图信息模型文件
18. 支持 tga 图像格式文件
19. 支持 svg 矢量图像格式文件
20. 支持 mp3, wav, mp4, flv 等音视频格式文件
21. 支持 avi, mov, rm, webm, ts, rm, mkv, mpeg, ogg, mpg, rmvb, wmv, 3gp, ts, swf 等视频格式转码预览

更多介绍请参考 [kkFileView 概述](#)

4. 部署过程

kkFileView 部署支持 Docker 容器环境部署以及发行包一键脚本启动两种方式。由于官方提供Docker 镜像发行包不是最新版（镜像版本为 4.1，最新版

是 4.4)，而且考虑后面可能需要修改部分代码和配置文件，所以采用自行构建部署的方式。

4.1. 拉取代码

```
git clone https://gitee.com/kekingcn/file-online-preview.git
```

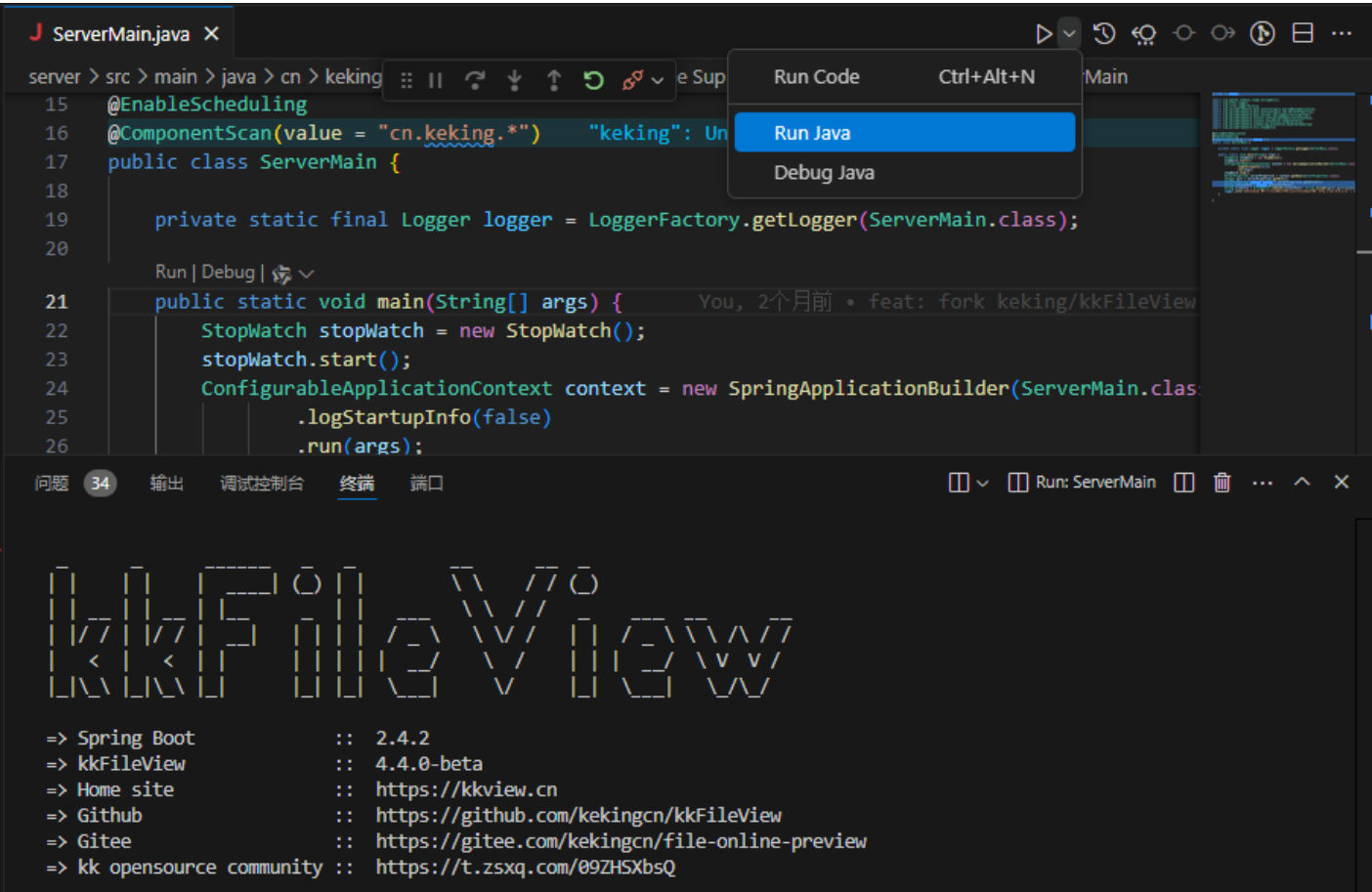
注意默认拉取的代码需要是 master 主分支，需要切换到 4.4.0-beta 的 tag 版本。

内部仓库地址: [yz-server/kkfileview](#)

4.2. 本地运行

由于前端习惯用的是 VSCode，而项目是 spring boot 技术，需要安装 java 环境，具体操作就不展开了，可以参考一下[VSCode搭建java编译环境、使用VSCode实现Java项目管理...](#)

找到 server\src\main\java\cn\keking\ServerMain.java 主入口，在 VSCode 中执行 `Run Java` 命令



看到启动成功后，浏览器访问 <http://127.0.0.1:8012/> 即可访问预览系统。

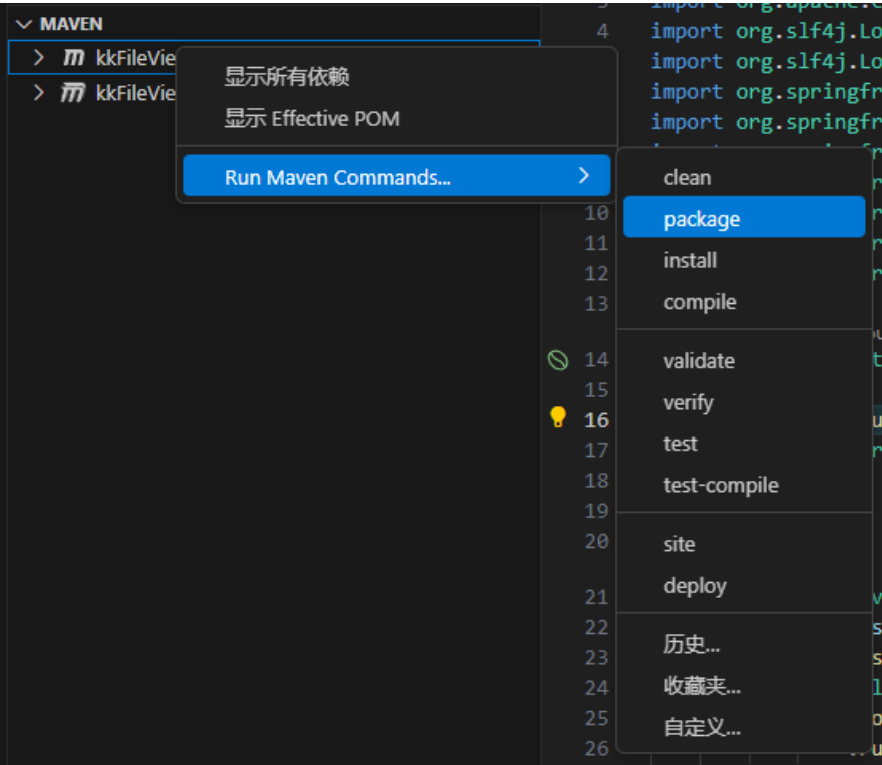
4.3. 代码修改

- 对系统文件的图标修改替换成公司的图标
- 对预览页面中一些关键字进行处理
- 对预览页面标题统一调整为“文件预览”，避免部分文件文件名获取失败显示异常问题
- 对预览失败和提示等页面的文案提示和图片展示进行修改，统一风格
- 修复部分页面缺省 commonHeader.ftl
- 添加 maven-surefire-plugin 插件，跳过测试用例
- 修改配置文件 /server/src/main/config/application.properties（根路径设置为相对路径，避免前端项目使用代理转发进行访问时，文件预览系统的资源加载失败）

```
# base.url = ${KK_BASE_URL:default}
base.url = .
```

4.4. 打包构建

执行 maven 的 `package` 命令，进行打包构建



构建完成后，就可以在 `/server/target` 目录下找到 `kkFileView-4.4.0-beta.tar.gz` 部署包。如果后续代码更新迭代，可以直接上传 `kkFileView-4.4.0-beta.jar` 覆盖源文件，然后重启服务即可。

4.5. 部署

1. 将 `kkFileView-4.4.0-beta.tar.gz` 拷贝到服务器进行解压
2. 打开解压后文件夹的 `bin` 目录（`server/src/main/bin`），运行 `startup.sh` 脚本（使用 `root` 用户运行 - `sudo su`）
3. 浏览器访问服务器 `8012` 端口，即可看到项目演示首页

正式环境部署中没有开放这个演示页面，访问都是只能通过内网进行转发，避免暴露项目被恶意使用或攻击

5. 踩坑

遇事不要慌，先看看官方的[常见问题](#)能不能解决~

这里记录下这次踩过的坑：

5.1. 找不到office组件，请确认'office.home'配置是否有误

部署到服务器执行启动脚本后，发现运行失败了，执行 `showlog.sh` 的脚本查看日志，提示“找不到office组件，请确认'office.home'配置是否有误”，查看了安装脚本，发现也有 `office` 组件的安装，也提示了安装成功，但是就是提示找不到，于是查找对应目录确实也没有安装到。于是进行单独安装了，也没效果，查看了网上的踩坑贴和 `issue`，推断出是系统自带的 `libreoffice` 冲突问题，需要先卸载再进行安装。类似 `sudo yum remove $(sudo yum list installed | grep libreoffice | awk '{print $1}')` 命令进行移除，然后重新执行启动脚本（里面包括了安装脚本）。

另外如果安装的目录不是配置文件中默认的 `/opt/libreoffice7.5` 目录，还需要修改配置文件参数，修改后注意重启应用，或者可以通过软链接的形式处理。

5.2. 项目根路径配置问题

正如前面所说，这预览系统只能通过内网访问，不能直接外网访问，所以使用默认的根路径 `base.url = ${KK_BASE_URL:default}` 配置时会导致几种情况：

部分页面静态资源访问路径错误

部分文件资源地址是 **https** 会导致提示证书问题

转发到内网域名映射时，加载资源会出现超时情况

官方文档上指出，一般使用 **nginx** 代理出现预览异常的，建议 **base.url** 改成部署所在的域名访问。

实际上改成域名，转发的页面路径是正常的，而预览页面资源请求的根路径会强制固定为配置的域名，导致 **nginx** 转发匹配上不对，静态资源又会访问不到，导致加载失败。

解决方案是使用的相对路径 `"."`，**base.url = .** 让预览页面发出的请求都相对于当前访问路径，相当于所有的资源又会走 **nginx** 代理转发到文件预览系统上。

不过这样会导致直接访问文件预览系统会找不到页面，但实际上预览系统的演示和接入说明页面并不需要在正式环境上使用。