链滴

Vue+SpringBoot 在 docker 下操作实践：上传 doc 转 PDF 并让前端同步获取转换结果

作者：liumapp

原文链接：https://ld246.com/article/1531145096462

来源网站：链滴

Doc文档的批量上传并转换为PDF文件，前端同步获取转换结果并提供下载

Github项目源码: liumapp/synchronizing-doc-convert-results

功能介绍

- 前端用户上传一个或者多个doc/docx文档
- 后端异步执行对文档的转换操作，利用了rabbitmq队列
- 前端可以同步获取转换结果，并以表格形式提供下载操作

介绍动图您可以在Github项目文档中浏览

如何使用

Docker

首先，请确保您的本地系统中有docker、docker-compose及maven的支持

其次，请确保您没有修改过synch-service的项目配置文件，如果修改过，请确保spring.profiles.activ的值为docker，并且与synch-rabbitmq容器的连接信息与docker-compose.yml的保持一致

安装镜像

拷贝项目到本地后，执行脚本build-image.sh来安装镜像

该镜像的安装，是由docker-maven-plugin驱动的，所以需要您确保在执行脚本前，本机提供maven版本以上的支持

启动容器

利用docker-compose，执行命令:

```
docker-compose up -d
```

来启动容器

启动后的视图您可以Github项目文档中浏览

之所以需要用到docker-compose进行容器的编排操作，是因为系统依赖于rabbitmq的支持

rabbitmq本身的存在，是以一个独立的容器来运行，并通过配置docker-compose，使两者共享一网关来进行数据交互

网关配置如下:

```
networks:
  synchronizing-doc-convert-results:
    driver: bridge
```

并在两个容器的配置项中，添加以下内容:

```
networks:
  - synchronizing-doc-convert-results
```

停止容器

利用docker-compose，执行命令:

```
docker-compose down
```

来停止容器

需要注意的是，停止容器本身，并不会删除镜像文件

删除镜像

在项目根目录下，执行脚本rm-image.sh删除镜像文件，请注意，该操作的执行前提是容器处于停止态

IDEA

考虑到部分用户可能对Docker不是太了解，所以这个项目也可以使用传统的方式来运行

但是项目本身使用了前后端分离的形式，前端项目为synch-ui，采用vue2.0框架实现，后端项目为synh-service，采用springboot1.5.6框架实现

所以如果您希望同时运行前后端项目的话，需要您的系统同时具备Java环境和Nodejs环境的支持

但考虑到大部分开发人员并不会选择全栈作为自己的发展发现，所以为纯Java开发人员提供了一定的利性:

synch-service利用thymeleaf已经加载了synch-ui编译后的静态文件，并完成了相关配置，所以纯Jav开发人员只需要将synch-service导入IDEA（导入过程省略），并且修改application.yml配置文件的sring.profiles.active值为dev，在启动后访问 http://localhost:2020 也可以查看系统效果。

与此同时，如果您是一名纯前端开发人员，在webstorm中打开了synch-ui项目，并利用

```
npm run dev
```

启动项目后，您需要同时启动后端项目才能够在 http://localhost:8080 查看系统效果。

如果您对前端代码做了一定的修改，而又希望将最新的效果导入synch-service中，那么请您执行脚本pdate-ui.sh自动完成synch-ui编译文件的导入工作。