2017年2月20日

你的

word转pdf并建网站，参考当下手机上编辑能分享到朋友圈上的功能——设置展示在厨框的文字和图片

* Word自动转成html5，便于浏览器、手机预览（半天4小时）
* Ppt自动转成html5，便于浏览器、手机预览（半天3小时）
* 建网站能够上传word，以html形式预览（半天4小时）
* 设置注册，登录系统（半天4小时）

重点性能：

* 内容浏览体验好——转换后在电脑、手机上浏览方便、清楚
* 上传、登录方便
* 界面简洁大方（参考微信、朋友圈的界面）

2017年2月22日

前几天试了一下github上将word转换成html的例程，发现排版都有些问题。今天只能尝试使用LibreOffice的命令来实现转换了。

LibreOfffice命令行：

soffice.exe -headless -invisible -convert-to pdf myfile.docx

下面这条命令可以用：

**soffice --headless --invisible --convert-to html:"XHTML Writer File" word转html文档.docx --outdir ~/Desktop**

<https://ask.libreoffice.org/en/question/47252/convert-to-html-creates-terribly-invalid-html/>

2017年2月22日下午：

先用 libreofffice的命令行实现word2html的转换，先把整个网站搭建起来

Manager功能：

* 能够上传word文档
* 转换word文档并保存至另一个文件夹
* 预览转换后的html文件

User功能：

* 用户名、密码注册
* 文档方框式预览

LibreOffice SDK接入

2017年2月25日晚：

界面需要修改的地方：

* 登录界面“/”：去掉SSO ID 和 邮箱 选项，修改”密码错误“的class样式
* 注册界面“newuser”：去掉SSO ID 改成 “再次输入密码”，增加两次密码输错的提示
* 文档管理界面：去掉”用户列表“链接（待定）

增加cookie功能或是jssesionId

## 功能实现步骤：

**Eclipse下tomcat的server运行目录**

/home/freejoy/Software/javaStructure/spring/neonworkspace/.metadata/.plugins/org.eclipse.wst.server.core/tmp0/wtpwebapps/WordConvertWeb/WEB-INF/views/fileUpLoad

1. 实现文件类型过滤 ： <form:input type=*"file"* accept=*".docx"* path=*"file"* id=*"file"* class=*"form-control input-sm"*/>
2. 实现word转换

创建目录 new File(“路径”).mkdirs()

文件名中文

1. 界面中英文变换

邮箱不能为空

密码

1. 实现cookie登录认证

2017年2月23日下午：

1. Hibernate标注含义：<http://blog.csdn.net/dodo0_0/article/details/4557850>

2017年2月28日，朱总汇报：

朱总，上周采用springmvc框架和mysql数据库搭建了word转html网站，网址为：http://106.39.137.162:8080/wordconvert/

当前已实现的功能：

1. 用户注册、登录
2. 用户以html方式预览上传的word文件
3. 管理员上传word文档
4. 管理员管理用户权限

存在的问题：

1. word转html采用的是调用系统命令行的方式，为此需要系统安装LibreOffice开源office软件

下一步内容：

1. 探索采用LibreOffice SDK来进行word转html
2. 探索ppt转html方式

朱总，麻烦您看一下哪些地方需要现在修改不？

**2017年3月1日 word2html向windows平台移植**

Jdk下载地址： <http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>

Apacha-tamcat下载地址：

<http://tomcat.apache.org/download-70.cgi>

Phpmyadmin下载地址：

<https://www.phpmyadmin.net/downloads/>

Mysql下载地址：

<https://dev.mysql.com/downloads/file/?id=467269>

Php下载地址：

<http://windows.php.net/download/>

Jdk、tomcat在windows下的配置：

1. **jdk环境变量配置**

变量名为JAVA\_HOME，变量值为：C:\JDK（具体请根据你安装的JDK路径而定，如果是默认应该是这个形式C:\Program Files\Java\jdk1.7）

变量名为CLASSPATH，变量值为.;%JAVA\_HOME%\lib;

变量名为PATH，变量值为%JAVA\_HOME%\bin

至此完成了JDK的环境配置，请在cmd命令行界面下输入java -version测试是否安装成功。

1. **Tomcat环境变量配置**

解压下载文件包

新建TOMCAT\_HOME变量

变量名TOMCAT\_HOME

变量值c:\tomcat

变量值即为我们下载的tomcat解压路径，在这里我的是c:\tomcat，

变量名CATALINA\_HOME

变量值c:\tomcat

没错，CATALINA\_HOME的变量值与TOMCAT\_HOME的变量值是一样的。

在系统变量中找到Path变量名，双击或点击编辑，在末尾添加如下内容

;%CATALINA\_HOME%\bin;%CATALINA\_HOME%\lib

这里要注意，各个变量值之间一定要用;分隔。

在cmd命令窗口下输入startup回车，运行如下图所示

在浏览器中输入http://http://localhost:8080

1. **mysql配置**

<http://www.cnblogs.com/wangjunyan/p/5183366.html>

1、我的电脑->属性->高级->环境变量->path

如:C:\Program Files\MySQL\MySQL Server 5.6\bin

2、修改my-default.ini

在其中修改或添加配置：

[mysqld]

basedir=C:\Program Files\MySQL\MySQL Server 5.6（mysql所在目录）

datadir=C:\Program Files\MySQL\MySQL Server 5.6\data （mysql所在目录\data）

3.以管理员身份运行cmd（win10右键左下角开始按钮选择以管理员身份运行cmd即可）

以管理员身份运行cmd（一定要用管理员身份运行，不然权限不够），

输入：cd C:\Program Files\MySQL\MySQL Server 5.6\bin 进入mysql的bin文件夹(不管有没有配置过环境变量，也要进入bin文件夹，否则之后启动服务仍然会报错误2)

输入mysqld -install(如果不用管理员身份运行，将会因为权限不够而出现错误：Install/Remove of the Service Denied!)

安装成功

4.运行mysqld --initialize（标题问题所在，若没有init则不存在data目录，自然无法启动成功）

5.安装成功后就要启动服务了，继续在cmd中输入:net start mysql,服务启动成功！

服务启动成功之后，就可以登录了，输入mysql -u root -p（第一次登录没有密码，直接按回车过）,登录成功！

**删除data文件夹，报“另一程序打开文件”解决方法：**

（右键）Windows 7任务栏——启动任务管理器——性能——资源监视器——CPU选项卡——关联的句柄——搜索句柄——（输入）要删除的文件夹名——搜索到与文件夹名句柄相关联的进程（由于此程序进程正在调用文件夹，才造成了对该文件夹删除的失败。）——(右键)该进程——结束进程——弹出警告对话框——确认后即可结束该进程，然后就可以完美删除该文件夹了

**Mysql在windows下utf8设置：**

<http://blog.csdn.net/u013474104/article/details/52486880>

在mysql安装目录下，将 my-default.ini 复制一分保存为 my.ini

做如下修改：

[mysqld]

character-set-server=utf8

collation-server=utf8\_general\_ci

[mysql]

default-character-set = utf8

[mysql.server]

default-character-set = utf8

[mysqld\_safe]

default-character-set = utf8

[client]

default-character-set = utf8

## **4、Tomcat启动、停止命令**

<http://blog.csdn.net/bas_ball/article/details/52343802>

shutdown:运行catalina stop以停止tomcat的运行

startup:运行catalina start以启动tomcat运行

**5、Windows下tomcat配置站点：** <http://blog.csdn.net/wy818/article/details/7240294>

<host>下增加

<Context docBase="D:\pafalearning\userapp\dist\tomcat\userapp.war" path="/userapp" reloadable="true"/>

1. **apache在windows下的配置**

修改Apache24/conf/httpd.conf文件：

1. Define SRVROOT为安装目录
2. 修改Listen 80 改为Listen 9090 （80端口被IIS占用）

7、apache、mysql、php、phpmyadmin在windows下的配置：

<http://jingyan.baidu.com/article/46650658302853f548e5f861.html>

1）下载Apache，请直接登录https://www.apachehaus.com/cgi-bin/download.plx并根据操作系统位数下载32位或64位。

2）找到E:\ampServer\Apache24\conf\httpd.conf文件，最好用编辑器打开，有行号，找到：Define SRVROOT 如图修改为你Apache解压存放的路径

打开cmd窗口（win+R），输入："E:\ampServer\Apache24\bin\httpd.exe" -k install -n apache24 复制时注意引号。然后回车，如图显示，即为主服务安装成功。

打开浏览器，访问http://localhost 若如图显示即为Apache安装成功。

3）下载PHP，请直接登录 http://windows.php.net/download#php-7.0并根据操作系统位数下载32位或64位。

找到E:\ampServer\php-7.0.10\php.ini-development文件，重命名为php.ini，打开后，找到extension\_dir = “ext”，去掉前面的分号

找到E:\ampServer\Apache24\conf\httpd.conf文件，打开后在如图显示的位置加入代码：

#\*\*\*\*\*\*\*\*\*\*\*php与Apache整合\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#让apache载入php处理模块

LoadModule php7\_module E:/ampServer/php-7.0.10/php7apache2\_4.dll

#指定php的ini文件，该文件是对php的一些配置

PHPIniDir "E:/ampServer/php-7.0.10"

AddType application/x-httpd-php .php .phtml

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

在E:\ampServer\Apache24\htdocs目录下，新建test.php文件内容如下：

<?php

phpinfo();

?>

然后，在浏览器访问http://localhost/test.php 若如图显示即为php安装整合成功。

4）下载MySQL，请直接登录 http://dev.mysql.com/downloads/mysql/5.7.html并根据操作系统位数下载32位或64位

在E:\ampServer\mysql-5.7.14目录下，新建data文件夹，然后配置环境变量，将E:\ampServer\mysql-5.7.14\bin；加入到path中。接着，打开cmd窗口（win+R），输入：mysqld -install然后回车，如图显示，即为安装成功。启动mysql服务，继续在cmd中输入:net start mysql，回车。继续在cmd中输入:mysql -uroot -p回车。因为密码为空，所以直接回车即登录成功。继续在cmd中输入:SET PASSWORD FOR 'root'@'localhost' = PASSWORD('新密码');回车。密码设置成功。

mysql的整合，php.ini去掉extension=php\_mbstring.dll 和 extension=php\_mysqli.dll前面的分号。然后，在E:\ampServer\Apache24\htdocs目录下，新建testsql.php文件内容如下：

<?php

$coun=mysqli\_connect("localhost","root","之前设的新密码（数据库密码）");

if ($coun) {

echo"连接成功";

}else{

echo "连接失败";

}

?>

然后，在浏览器访问http://localhost/testsql.php 若显示“连接成功”即为mysql安装整合成功。

1. 下载phpMyAdmin，请直接登录https://www.phpmyadmin.net/downloads/，并解压到Apacha24/htdocs下

E:\ampServer\Apache24\htdocs\phpMyAdmin-4.6.4\libraries\config.default.php文件，打开后配置$cfg['Servers'][$i]['password'] = '新密码';然后设置短语密码$cfg['blowfish\_secret'] = '';如果不知短语密码想设置成什么，以后设置也行。

在浏览器访问http://localhost/phpMyAdmin-4.6.4/index.php进入登录页面输入用户名：root密码：之前设的新密码（数据库密码）登录即可

## 遇到的问题：

1. **@Configuration @Component 和 @Bean的区别**

@Configuration最大，它用于把所有的Bean、Component组合起来，供程序调用，相当于ApplicationContext调用.xml的效果

@Component用于组织类，可以通过@Autowired方式实现自动组装，@Service @Repository都属于@Component

@Bean用于定义单例模式的类

1. **PropertyPlaceholderConfigurer中 ClassPathResource 与 FilePathResource http://snkcxy.iteye.com/blog/1913796**

propertyPlaceholder.setLocation(**new** ClassPathResource("fileupload.properties"));

1. Properites <http://websystique.com/spring/spring-propertysource-value-annotations-example/>

### 4）Jsp、mysql中文乱码问题

<http://blog.csdn.net/qq_32144341/article/details/51318390>

**搜索：“ubuntu mysql 5.7 character set”，搜索一定要精确，细化！！！**

|  |
| --- |
| 1、在终端中输入 sudo gedit /etc/mysql/mysql.conf.d/mysqld.cnf 命令  2、打开mysqld.cnf 文件，在lc-messages-dir  = /usr/share/mysql 语句后添加 character-set-server=utf8 语句   1. 在终端输入 sudo gedit /etc/mysql/conf.d/mysql.cnf 命令打开mysql.cnf配置文件，如图添加代码：default-character-set=utf8 如图： 2. 在终端中输入 /etc/init.d/mysql restart 命令重启MySQL服务，如图所示，重启成功。 |

**Mysql建表：**

|  |
| --- |
| CREATE TABLE `APP\_USER` (  `id` int(11) NOT NULL AUTO\_INCREMENT,  `EMAIL` varchar(255) NOT NULL,  `ISADMIN` bit(1) NOT NULL,  `NAME` varchar(255) NOT NULL,  `PASSWORD` varchar(255) NOT NULL,  `PASSWORDCONFIRM` varchar(255) NOT NULL,  PRIMARY KEY (`id`),  UNIQUE KEY `UK\_7fwo1lohkm7ktmye0be5koglc` (`NAME`)  ) ENGINE=InnoDB AUTO\_INCREMENT=7 DEFAULT CHARSET=utf8 |
| CREATE TABLE `USER\_DOCUMENT` (  `id` int(11) NOT NULL AUTO\_INCREMENT,  `content` longblob NOT NULL,  `description` varchar(255) DEFAULT NULL,  `name` varchar(100) NOT NULL,  `type` varchar(100) NOT NULL,  `USER\_ID` int(11) NOT NULL,  PRIMARY KEY (`id`),  KEY `FK\_lcfjscnnac0sqjjevhlk7b7rd` (`USER\_ID`),  CONSTRAINT `FK\_lcfjscnnac0sqjjevhlk7b7rd` FOREIGN KEY (`USER\_ID`) REFERENCES `APP\_USER` (`id`)  ) ENGINE=InnoDB AUTO\_INCREMENT=5 DEFAULT CHARSET=utf8 |

**Mysql命令行：**

<http://www.cnblogs.com/shootercheng/p/5836657.html>

java中的常用编码UTF-8;GBK;GB2312;ISO-8859-1;   
对应mysql数据库中的编码utf8;gbk;gb2312;latin1

springMVC的Filter定义

|  |
| --- |
| @WebFilter(filterName = "Servlet3Filter", urlPatterns = "/\*")  **public** **class** FilterFileName **implements** Filter {  **protected** String encoding = **null**;  **protected** FilterConfig filterConfig = **null**;  **protected** **boolean** ignore = **true**;    **public** **void** destroy() {  System.***out***.println("过滤器销毁");  }  **public** **void** doFilter(ServletRequest request, ServletResponse response, FilterChain chain)  **throws** IOException, ServletException {  System.***out***.println("执行过滤操作");  **try**  {  HttpServletRequest httpRequest = (HttpServletRequest)request;  String method = httpRequest.getMethod().toLowerCase();  **if**(method.equals("post"))  {  //如果是post，即表单方法，直接设置charset即可  request.setCharacterEncoding("UTF-8");  response.setContentType("text/html;charset=UTF-8");  }  /\*else if(method.equals("get"))  {  //如果是get方法  request.setCharacterEncoding("UTF-8");  request = new HttpServletRequestWrapper((HttpServletRequest)request)  {  public String getParameter(String str)  {  try  {  return new String(super.getParameter(str).getBytes("iso-8859-1"),"GBK");  }  catch(Exception e)  {  return null;  }    }  };  }\*/    chain.doFilter(request, response);  }  **catch**(Exception e){}  }  **public** **void** init(FilterConfig config) **throws** ServletException {  System.***out***.println("过滤器初始化");    }  } |

### 5）docx下载并转换为html

|  |
| --- |
| // 获得文件名:  String realFileName = document.getName();  // 获取路径  String ctxPath = request.getSession().getServletContext().getRealPath(  "/")  + "/WEB-INF/views/fileUpLoad/";  File fileUpLoadDir = **new** File(ctxPath);  **if**(fileUpLoadDir.exists() == **false**)  {  fileUpLoadDir.mkdirs();  }  //File uploadFileDir = new File(ctxPath);  File uploadFile = **new** File(ctxPath + realFileName);  **if**(!uploadFile.exists())  {  System.***out***.println("word file does not exist");  FileCopyUtils.*copy*(document.getContent(), uploadFile);  }  String jspFileName = uploadFile.getAbsolutePath().replace(".docx", ".jsp");  **if**(**new** File(jspFileName).exists() == **false**){  System.***out***.println("html file does not exist");  String command = "soffice --headless --invisible --convert-to html:\"XHTML Writer File\" " + uploadFile.getAbsolutePath() + " --outdir " + ctxPath;  runSystemCmd(command);  File htmlFileName = **new** File( jspFileName.replace(".jsp", ".html") );  htmlFileName.renameTo(**new** File(jspFileName));    } |

### soffice --invisible --headless -convert-to html:"XHTML Writer File" 2月9日变更后电子银行交易量报告(0214).docx soffice --invisible --headless --convert-to html:"XHTML Writer File" 中 文.docx 不支持“（）”，不支持“空格”

### Spring的html解析

<http://blog.csdn.net/huangxy10/article/details/8185108>

|  |
| --- |
| Document doc = Jsoup.*parse*(**new** File(jspFileName), "UTF-8");  System.***out***.println(doc.head()); |

1. Response生成html页面

|  |
| --- |
| **private** **void** generateHtmlFormDoc(HttpServletResponse response, Document doc) **throws** IOException{  response.setContentType("text/html");  response.setCharacterEncoding("UTF-8");  PrintWriter out = response.getWriter();  out.println("<html>");  out.println(doc.head());  out.println(doc.body());  out.println("</html>"); } |

1. Hibernate以utf8方式创建TABLE

|  |
| --- |
| **public** **class** MySQLDialectUTF **extends** MySQL5Dialect{  @Override  **public** String getTableTypeString() {  **return** " ENGINE=InnoDB DEFAULT CHARSET=utf8";  }  }  hibernate.dialect = com.websystique.springmvc.configuration.MySQLDialectUTF |

1. 新增Error提示

|  |
| --- |
| FieldError ssoError =new FieldError("user","ssoId",messageSource.getMessage("non.unique.ssoId", new String[]{user.getSsoId()}, Locale.getDefault()));  result.addError(ssoError);  return "registration"; |

1. 密码输入框，form::password不能用type=”text”属性

|  |
| --- |
| <form:password path=*"password"* id=*"password"* class=*"form-control input-sm"* />  <form:input type=*"text"* path=*"name"* id=*"name"* class=*"form-control input-sm"*/> |

1. Interceptor拦截器用于登录、session验证 <https://www.tianmaying.com/tutorial/spring-mvc-interceptor>

|  |
| --- |
| @Configuration  @EnableWebMvc  public class WebConfig extends WebMvcConfigurerAdapter {  @Override  public void addInterceptors(InterceptorRegistry registry) {  registry.addInterceptor(new LocaleInterceptor());  registry.addInterceptor(new ThemeInterceptor()).addPathPatterns("/\*\*").excludePathPatterns("/admin/\*\*");  registry.addInterceptor(new SecurityInterceptor()).addPathPatterns("/secure/\*");  }  } |