Atitit springboot 访问html 静态资源

目录

[1.1. 默认的静态资源的路径是什么？ resources,public,static， 1](#_Toc20897)

[1.2. 自定义路径 1](#_Toc2229)

[1.3. 自己实现读取使用filter 自设计web服务器原理与实现 2](#_Toc847)

[1.4. 访问绝对路径 3](#_Toc370)

## 默认的静态资源的路径是什么？ resources,public,static，

/MEAT-INF/resources, resources,public,static，即在没有任何额外配置的情况下，

寻找静态资源的路径就是以上路经

## 自定义路径

方法二：通过@Configuration配置

@Configuration

public class ApplicationConfig extends WebMvcConfigurerAdapter {

@Override

public void addResourceHandlers(ResourceHandlerRegistry registry) {

/\*\*

\* 如果我们将/xxxx/\*\* 修改为 /\*\* 与默认的相同时，则会覆盖系统的配置，可以多次使用 addResourceLocations 添加目录，

\* 优先级先添加的高于后添加的。

\*

\* 如果是/xxxx/\*\* 引用静态资源 加不加/xxxx/ 均可，因为系统默认配置（/\*\*）也会作用

\* 如果是/\*\* 会覆盖默认配置，应用addResourceLocations添加所有会用到的静态资源地址，系统默认不会再起作用

\*/

registry.addResourceHandler("/\*\*")

.addResourceLocations("classpath:/META-INF/resources/")

.addResourceLocations("classpath:/resources/")

.addResourceLocations("classpath:/static/")

.addResourceLocations("classpath:/public/");

registry.addResourceHandler("/we/\*\*").addResourceLocations("file:F:/preview/");

super.addResourceHandlers(registry);

}

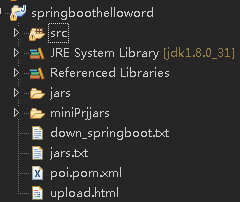
}

## 自己实现读取使用filter 自设计web服务器原理与实现

Atitit spirngboot 访问 html文件总结 自设计web服务器原理与实现

Url路由压力，读取url，获得项目更路径绝对路径，拼接为文件路径。读取文建内容输出即可

目录路径 upload。Html在项目跟目录



默认的要佳配置文件和放入指定目录。。麻烦放弃此种方法

使用springboot 拦截器测试，老是拦截不到uri ，都是/error

使用java的filter拦截，效果良好。。

<http://localhost:8080/upload.html>

|  |
| --- |
| ***@Component***  ***@WebFilter***(urlPatterns = "/\*", filterName = "authFilter")  **public** **class** **AuthFilter** **implements** Filter {  ***@Override***  **public** **void** **destroy**() {  // **TODO** Auto-generated method stub    }  ***@Override***  **public** **void** **doFilter**(ServletRequest arg0, ServletResponse arg1, FilterChain arg2)  **throws** **IOException**, **ServletException** {  HttpServletRequest **httpServletRequest**=(HttpServletRequest) arg0;  HttpServletResponse **httpServletResponse**= (HttpServletResponse) arg1;  **String** **uri**= httpServletRequest.getRequestURI();  **System**.***out***.println(uri);  **String** **url**= httpServletRequest.getRequestURL().toString();  **if**(uri.endsWith(".html"))  {  **String** **classFilePath** = **this**.getClass().getResource("/").getPath();  **String** **prjpath**=**new** File(classFilePath).getParent();  **String** **htmlpath**=prjpath+uri;  **byte**[] **s**=**FileUtils**.*readFileToByteArray*(**new** File(htmlpath));  **ServletOutputStream** **outputStream** = httpServletResponse.getOutputStream();  outputStream.write(s);  outputStream.flush();  }  } |

## 访问绝对路径

[http://localhost:8080/showhtml?path=D:\000000\index.html](http://localhost:8080/showhtml?path=D:/000000/index.html)

} **else** **if**( uri.equals("/showhtml"))

{

/// http://localhost:8080/showhtml?path=D:\000000\index.html

**String** **htmlpath**=httpServletRequest.getParameter("path");

outputHtml(httpServletResponse, htmlpath);

}**else**

arg2.doFilter(arg0, arg1);

}

**private** **void** **outputHtml**(HttpServletResponse httpServletResponse, **String** htmlpath) **throws** **IOException** {

**byte**[] **s** = **FileUtils**.*readFileToByteArray*(**new** File(htmlpath));

**ServletOutputStream** **outputStream** = httpServletResponse.getOutputStream();

outputStream.write(s);

outputStream.flush();

}