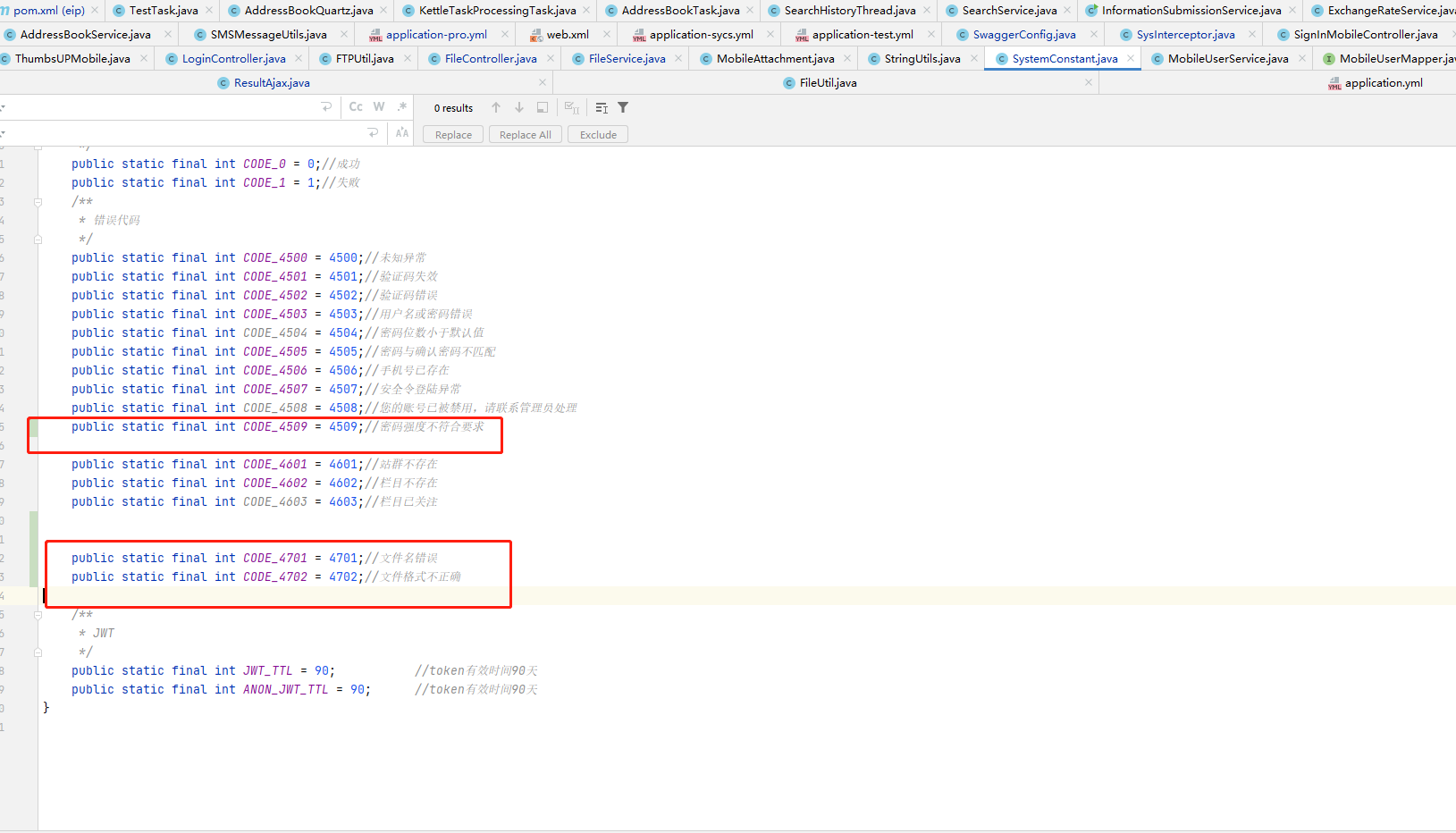
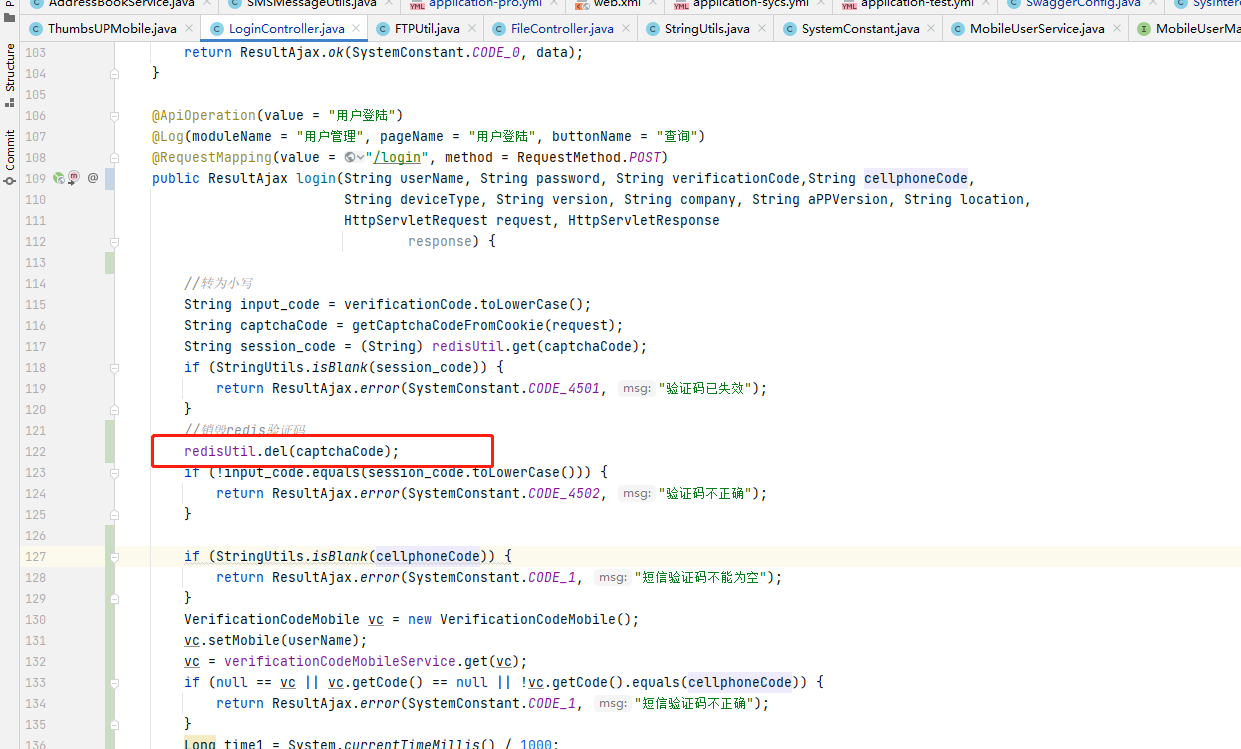
错误代码更新

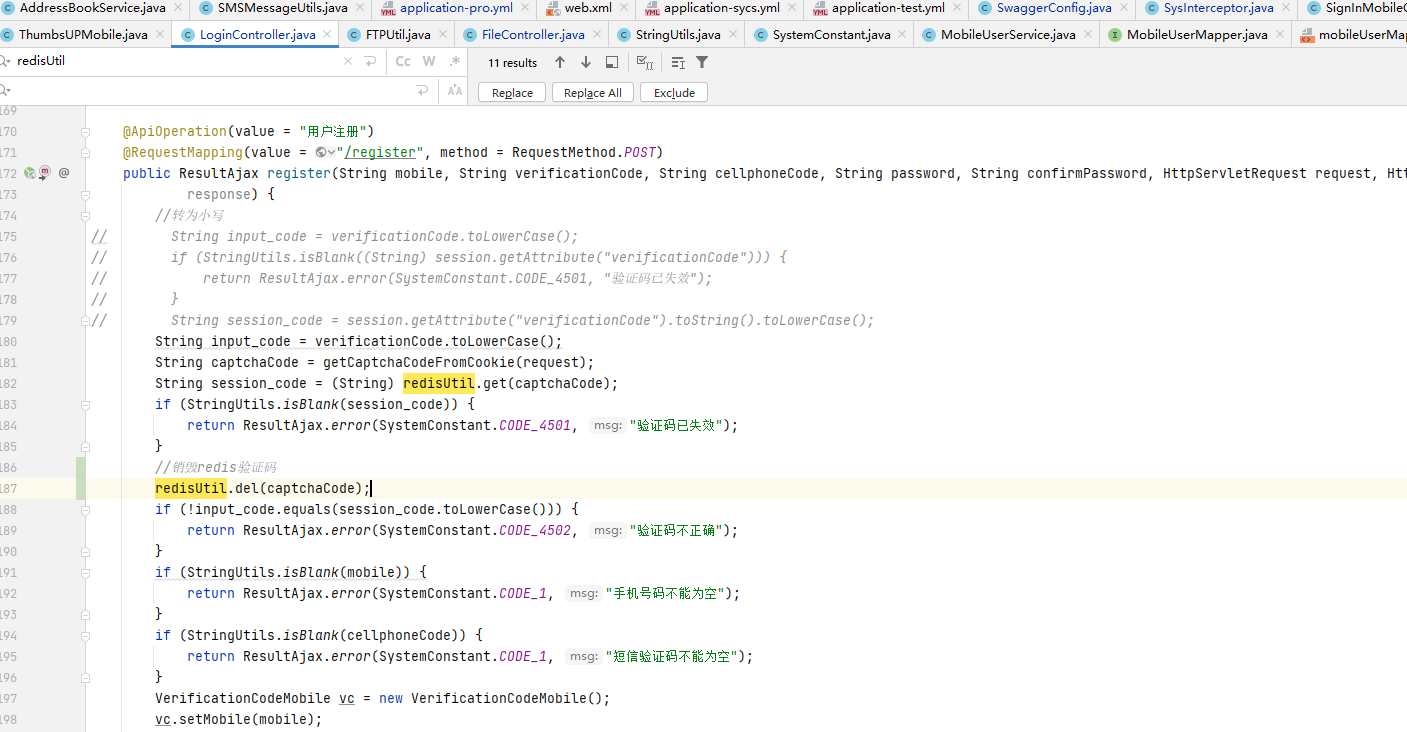


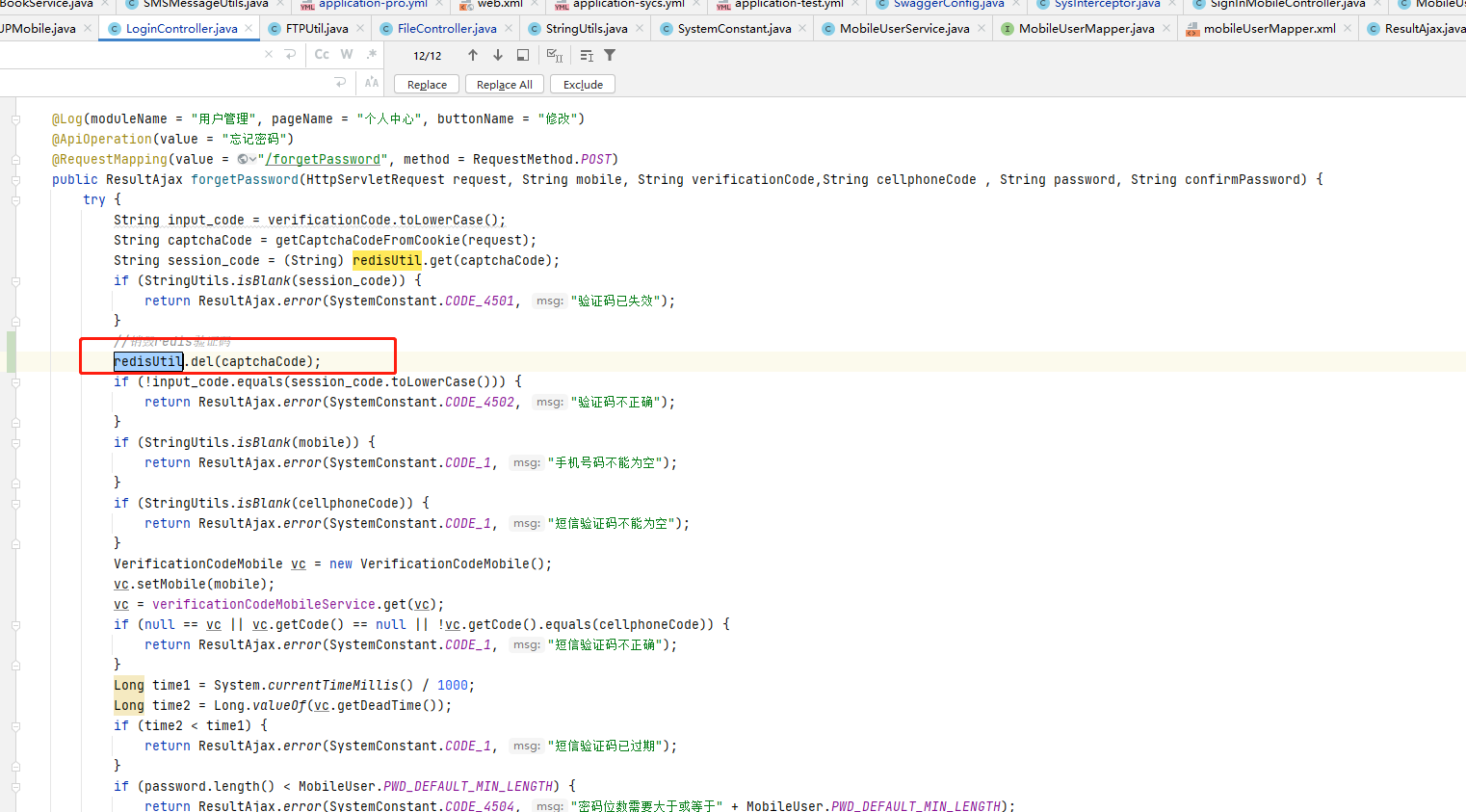
1. 验证码绕过(待验证)

验证码用过销毁(共三处, 前端登录,注册,忘记密码处需要添加验证失败刷新验证码)

redisUtil.del(captchaCode);







2. 弱口令

验证密码强度,共三处(前端注册,忘记密码,修改密码处需要添加密码强度验证

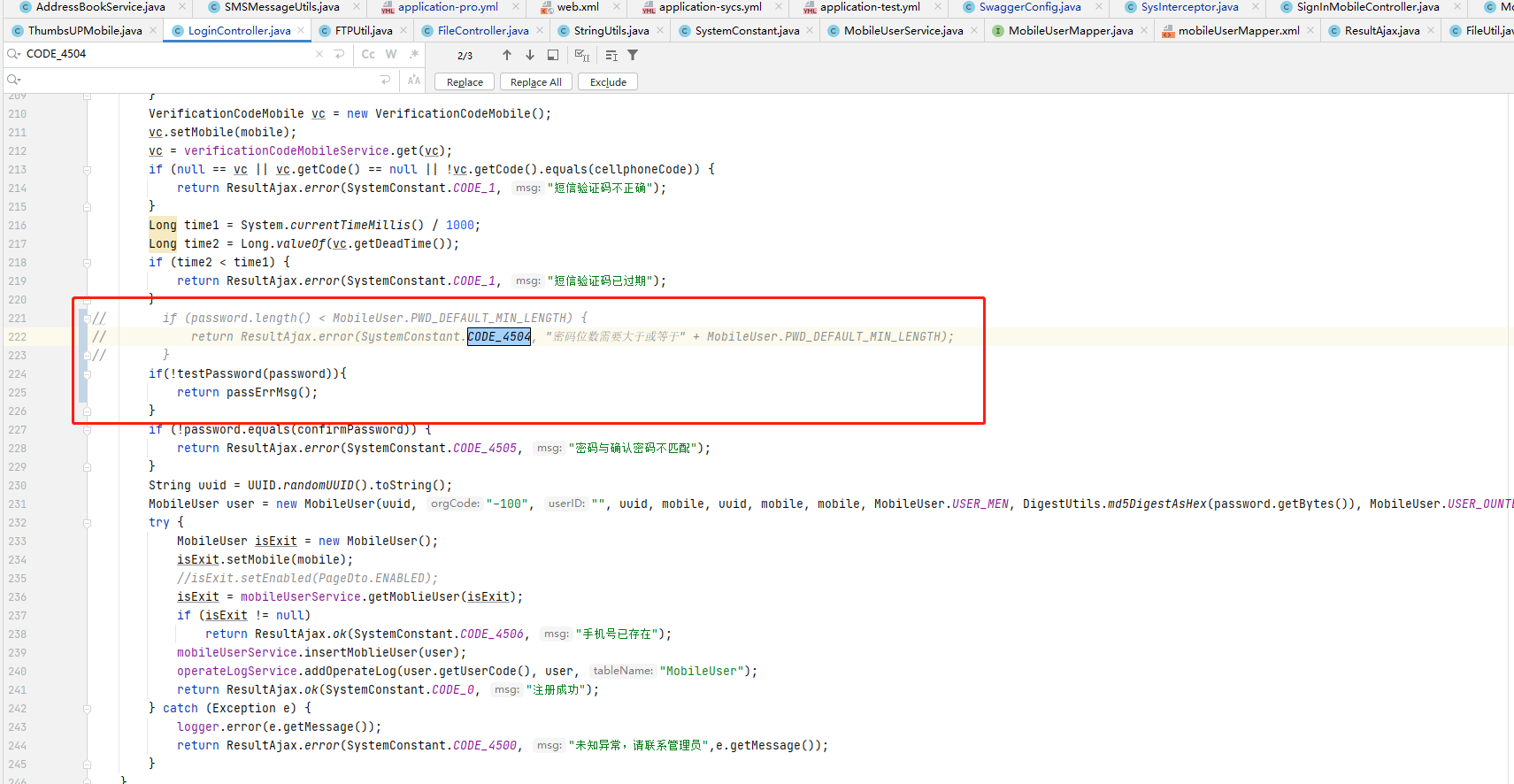
前端方法:

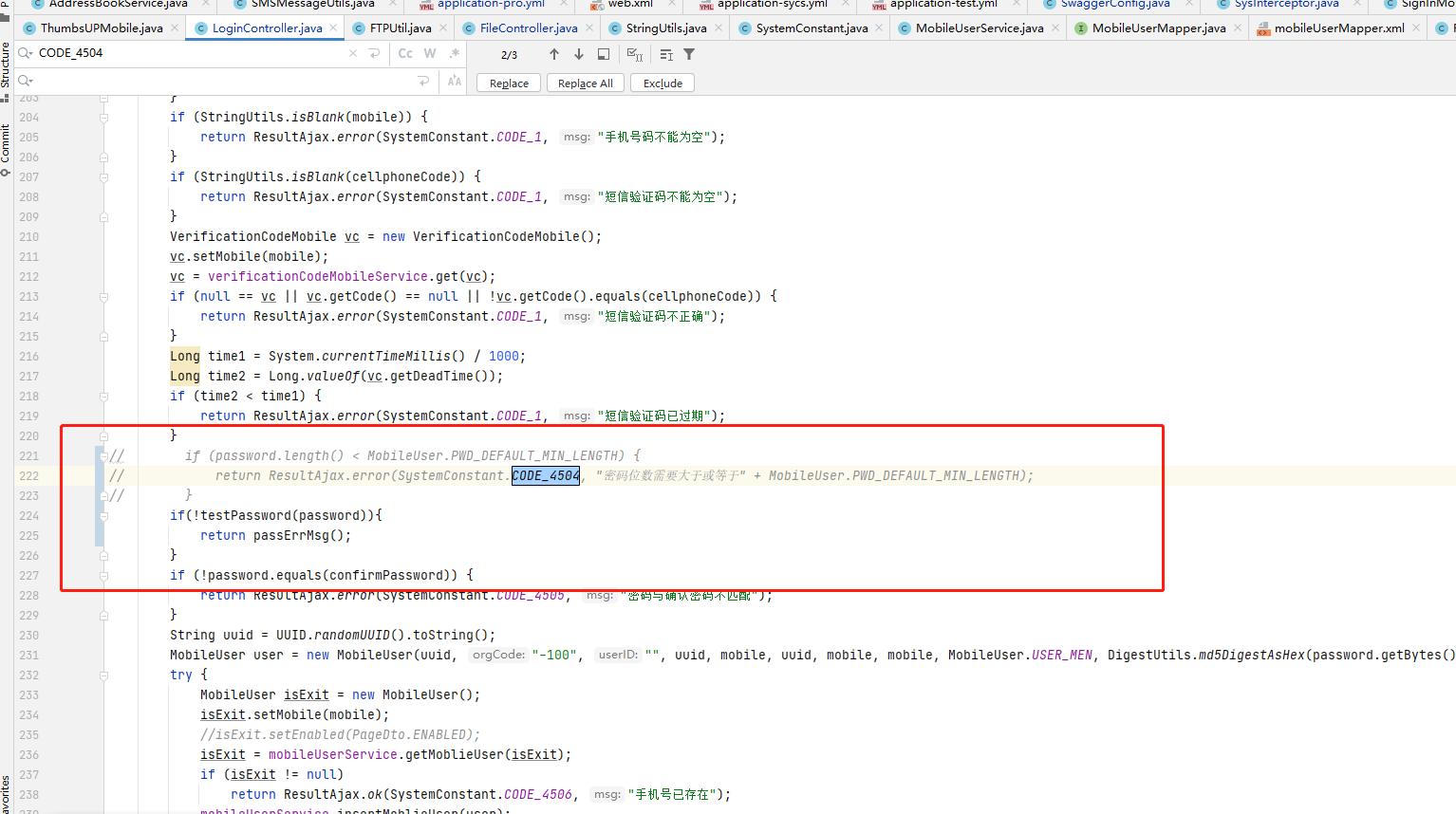
1. function checkoutPass2(pass){
2. var test = /^(?![a-zA-Z]+$)(?![A-Z0-9]+$)(?![a-z0-9]+$)(?![A-Z\\W\_!@#$%^&\*`~()-+=]+$)(?![a-z\\W\_!@#$%^&\*`~()-+=]+$)(?![0-9\\W\_!@#$%^&\*`~()-+=]+$)[a-zA-Z0-9\\W\_!@#$%^&\*`~()-+=]{8,16}$/;
3. **if**(!test.test(pass)) **return** '密码必须8到16位，包括数字、小写字母、大写字母、特殊符号中至少3类，且不能出现空格';
4. }

)

后台修改代码:

1. **if**(!testPassword(password)){
2. **return** passErrMsg();
3. }
4. **public** **boolean** testPassword(String password){
5. String regex = "^(?![a-zA-Z]+$)(?![A-Z0-9]+$)(?![a-z0-9]+$)(?![A-Z\\W\_!@#$%^&\*`~()-+=]+$)(?![a-z\\W\_!@#$%^&\*`~()-+=]+$)(?![0-9\\W\_!@#$%^&\*`~()-+=]+$)[a-zA-Z0-9\\W\_!@#$%^&\*`~()-+=]{8,16}$";
6. **if**(!password.matches(regex) || password.indexOf(" ")>=0){
7. **return** **false**;
8. }
9. **return** **true**;
10. }
12. **public** ResultAjax passErrMsg(){
13. **return** ResultAjax.error(SystemConstant.CODE\_4509, "密码必须8到16位，包括数字、小写字母、大写字母、特殊符号中至少3类，且不能出现空格");
14. }

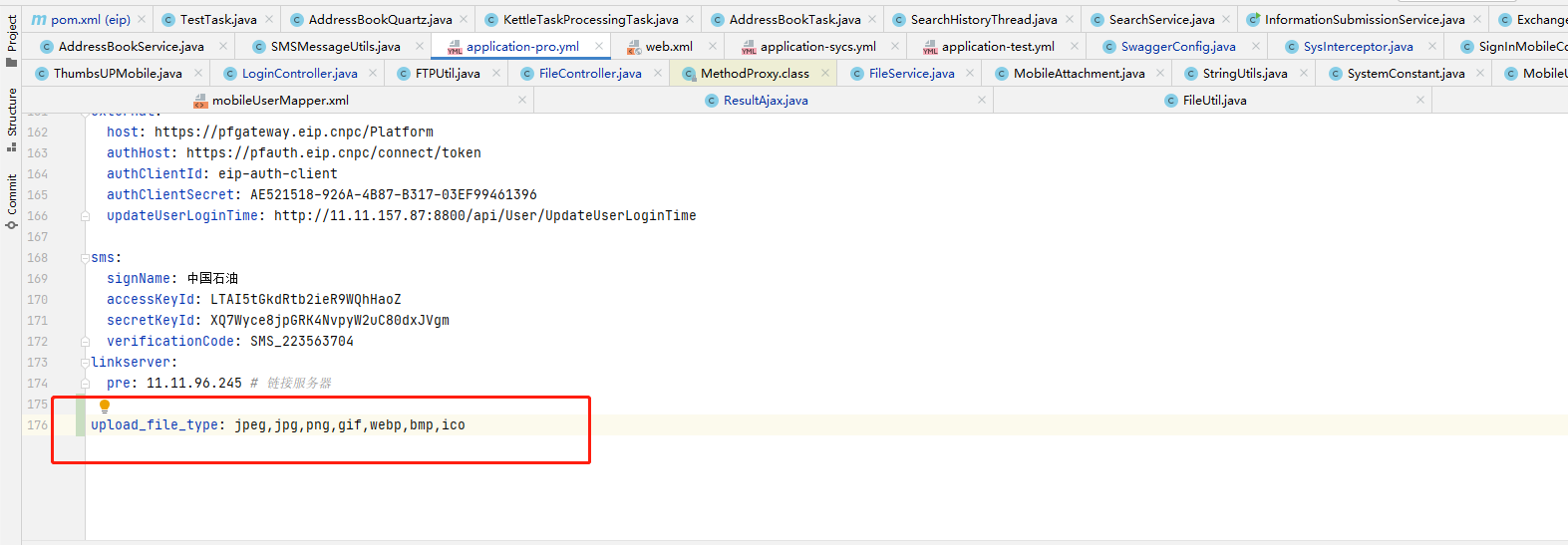


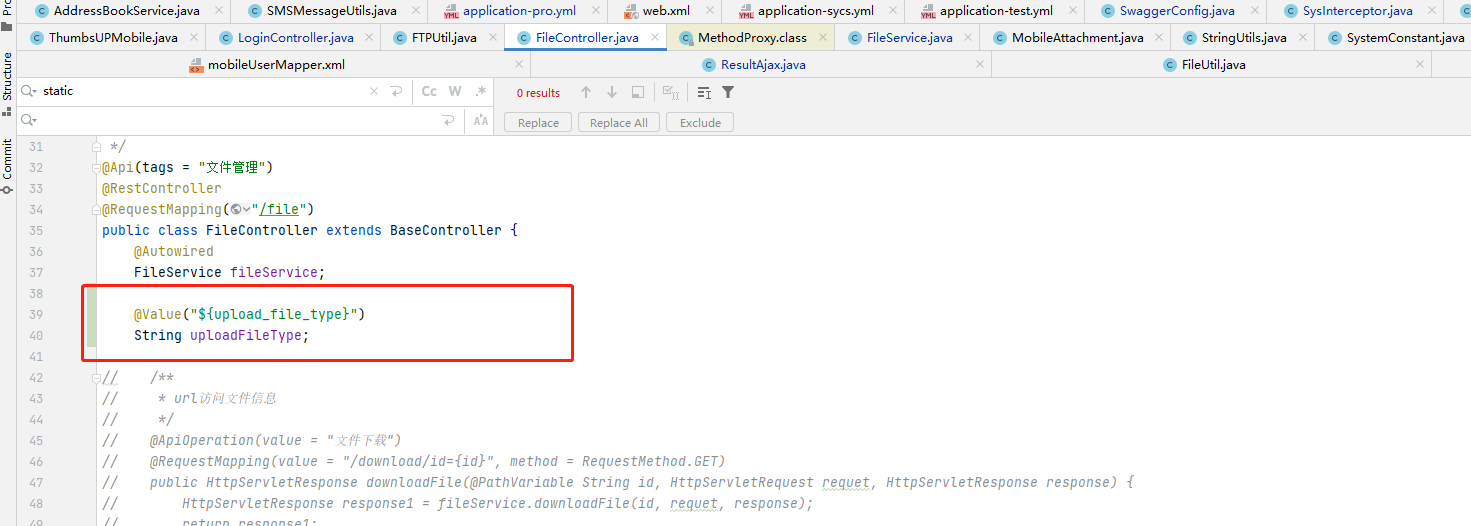




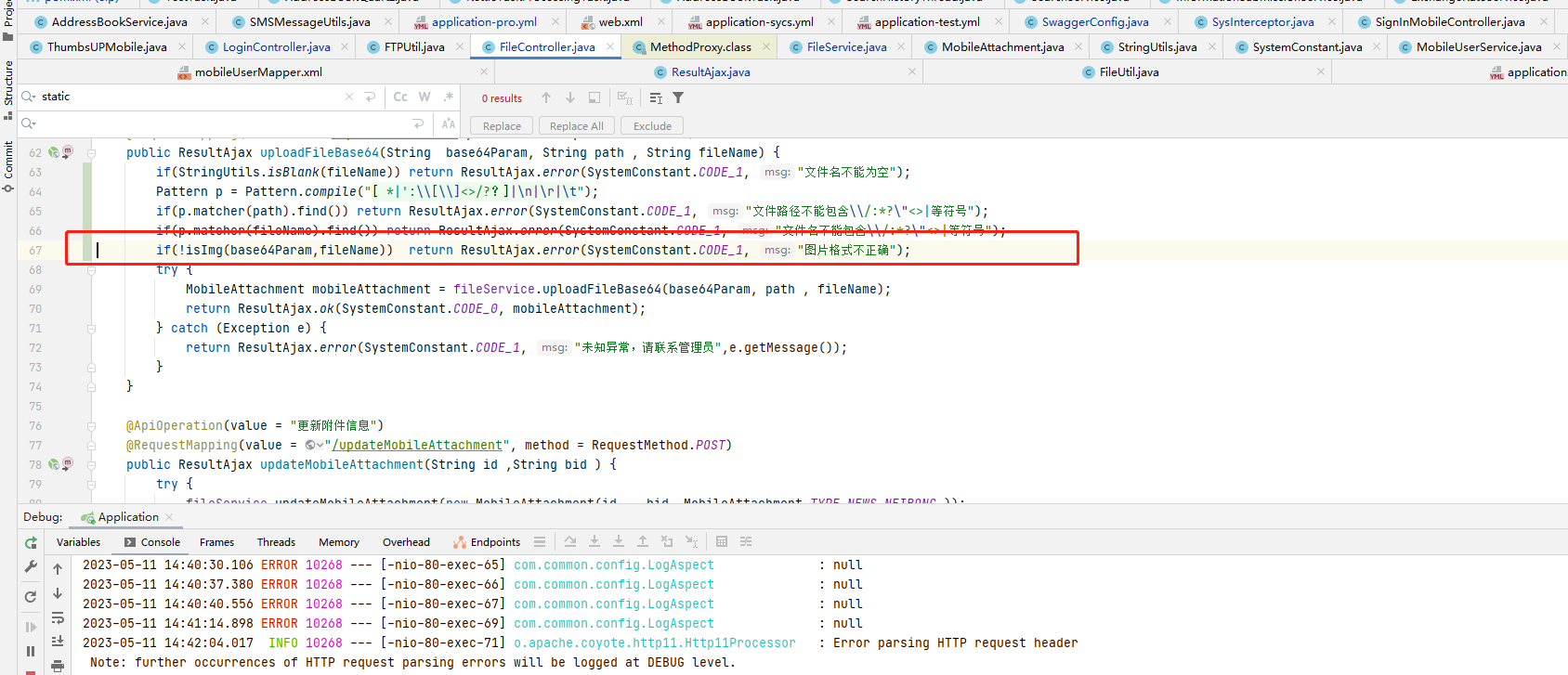
3. 任意文件上传

判断MIME Type, 文件后缀





1. **public** boolean isImg(String base64Param , String fileName) {
2. String type = fileName.substring(fileName.lastIndexOf(".")+1);
3. List<String> types = Arrays.asList(uploadFileType.split(","));
4. **if**(!types.contains(type.toLowerCase())){
5. **return** **false**;
6. }
7. byte[] buffer = **new** byte[0];
8. **try** {
9. buffer = **new** BASE64Decoder().decodeBuffer(base64Param);
10. } **catch** (IOException e) {
11. e.printStackTrace();
12. }
13. ByteArrayInputStream byteArrayInputStream = **new** ByteArrayInputStream(buffer);
14. **return** readType(byteArrayInputStream);
15. }
17. /\*\*
18. \* 输入流识别图片扩展名
19. \* @param is
20. \* @return
21. \* @throws IOException
22. \*/
23. **public** **boolean** readType(InputStream is){
25. **byte**[] bufHeaders = readInputStreamAt(is, 0, 8);
26. **if** (isJPEGHeader(bufHeaders)) {
27. **return** **true**;
28. }
29. **if** (isPNG(bufHeaders)) {
30. **return** **true**;
31. }
32. **if** (isGIF(bufHeaders)) {
34. **return** **true**;
35. }
36. **if** (isWEBP(bufHeaders)) {
37. **return** **true**;
38. }
39. **if** (isBMP(bufHeaders)) {
40. **return** **true**;
41. }
42. **if** (isICON(bufHeaders)) {
43. **return** **true**;
44. }
45. **return** **false**;
46. }
48. /\*\*
49. \* 标示一致性比较
50. \*
51. \* @param buf
52. \*            待检测标示
53. \* @param markBuf
54. \*            标识符字节数组
55. \* @return 返回false标示标示不匹配
56. \*/
57. **private** **boolean** compare(**byte**[] buf, **byte**[] markBuf) {
58. **for** (**int** i = 0; i < markBuf.length; i++) {
59. **byte** b = markBuf[i];
60. **byte** a = buf[i];
62. **if** (a != b) {
63. **return** **false**;
64. }
65. }
66. **return** **true**;
67. }
69. /\*\*
70. \*
71. \* @param fis
72. \*            输入流对象
73. \* @param skiplength
74. \*            跳过位置长度
75. \* @param length
76. \*            要读取的长度
77. \* @return 字节数组
78. \* @throws IOException
79. \*/
80. **private** **byte**[] readInputStreamAt(InputStream fis, **long** skiplength, **int** length){
81. **byte**[] buf = **new** **byte**[length];
82. **try** {
83. fis.skip(skiplength); //
84. fis.read(buf, 0, length);
85. } **catch** (IOException e) {
86. e.printStackTrace();
87. }
88. **return** buf;
89. }
91. // BMP图片文件的前两个字节
92. **private** **boolean** isBMP(**byte**[] buf) {
93. **byte**[] markBuf = "BM".getBytes();
94. **return** compare(buf, markBuf);
95. }
97. **private** **boolean** isICON(**byte**[] buf) {
98. **byte**[] markBuf = { 0, 0, 1, 0, 1, 0, 32, 32 };
99. **return** compare(buf, markBuf);
100. }
102. // WebP图片识别符
103. **private** **boolean** isWEBP(**byte**[] buf) {
104. **byte**[] markBuf = "RIFF".getBytes();
105. **return** compare(buf, markBuf);
106. }
108. **private** **boolean** isGIF(**byte**[] buf) {
109. // GIF识别符
110. **byte**[] markBuf = "GIF89a".getBytes();
111. **if** (compare(buf, markBuf)) {
112. **return** **true**;
113. }
114. // GIF识别符
115. markBuf = "GIF87a".getBytes();
116. **if** (compare(buf, markBuf)) {
117. **return** **true**;
118. }
119. **return** **false**;
120. }
122. // PNG识别符
123. **private** **boolean** isPNG(**byte**[] buf) {
124. **byte**[] markBuf = { (**byte**) 0x89, 0x50, 0x4E, 0x47, 0x0D, 0x0A, 0x1A, 0x0A };
125. // new String(buf).indexOf("PNG")>0 //也可以使用这种方式
126. **return** compare(buf, markBuf);
127. }
129. // JPEG开始符
130. **private** **boolean** isJPEGHeader(**byte**[] buf) {
131. **byte**[] markBuf = { (**byte**) 0xff, (**byte**) 0xd8 };
132. **return** compare(buf, markBuf);
133. }
135. // JPEG结束符
136. **private** **boolean** isJPEGFooter(**byte**[] buf) {
137. **byte**[] markBuf = { (**byte**) 0xff, (**byte**) 0xd9 };
138. **return** compare(buf, markBuf);
139. }

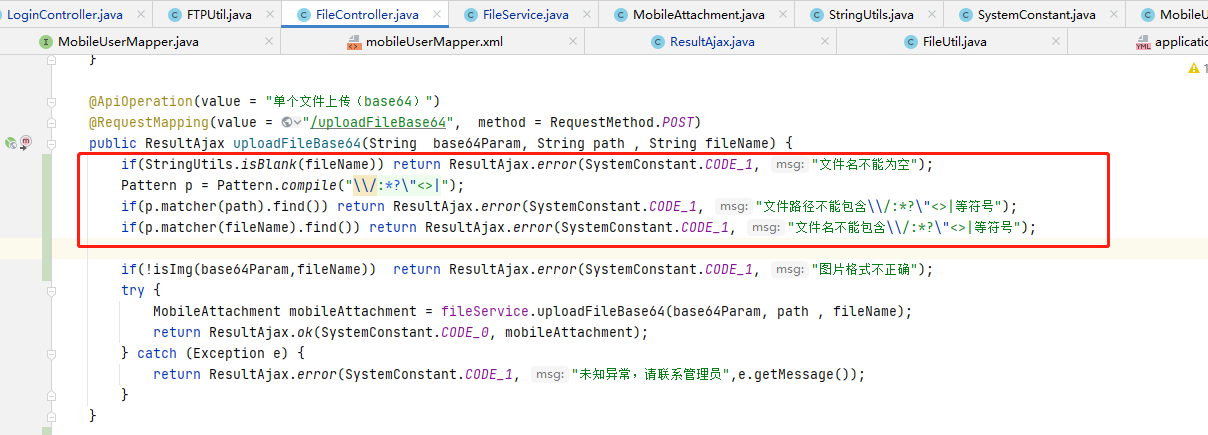


1. **if**(!isImg(base64Param,fileName))
2. **return** ResultAjax.error(SystemConstant.CODE\_4072, "图片格式不正确");

4. 跨目录上传

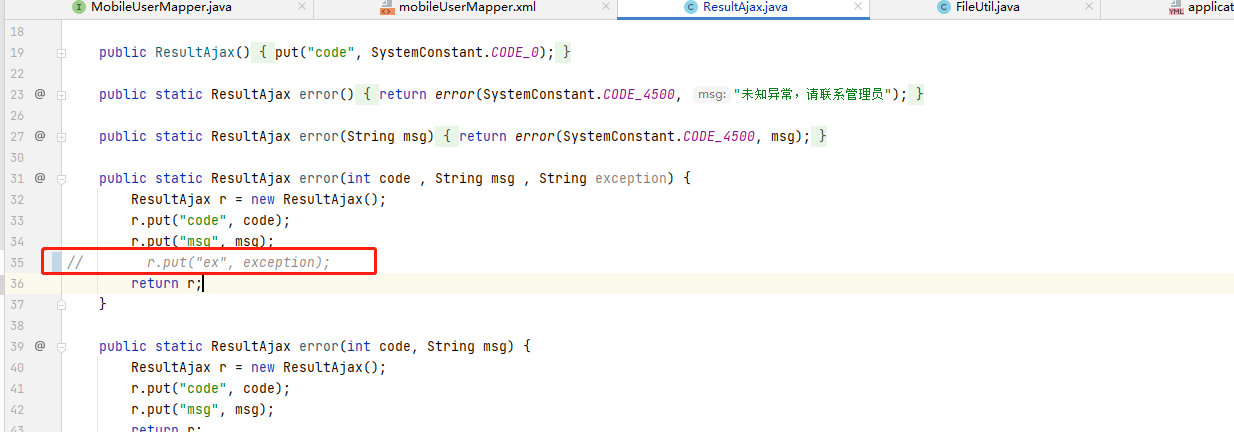
限制文件名称不能有特殊符号

1. **if**(StringUtils.isBlank(fileName)) **return** ResultAjax.error(SystemConstant.CODE\_4071, "文件名不能为空");
2. Pattern p = Pattern.compile("[ \*|':\\[\\]<>/?？]|\n|\r|\t");
3. **if**(p.matcher(path).find()) **return** ResultAjax.error(SystemConstant.4071, "文件路径不能包含\\/:\*?\"<>|等符号");
4. **if**(p.matcher(fileName).find()) **return** ResultAjax.error(SystemConstant.4071, "文件名不能包含\\/:\*?\"<>|等符号");



5. 敏感信息泄露

去掉返回详细错误信息(ex)



6. Druid未授权访问

配置文件关闭页面

druid:  
 stat-view-servlet:  
 enabled: false *# 添加此行配置即可关闭Druid管理界面*

