

```

1 package jp.silvercat.util;
2
3 import static org.junit.Assert.fail;
4
24 public class PdfUtilTest {
25     /**
26      * Log instance.
27      */
28     private static final Log LOG = LogFactory.getLog(PdfUtilTest.class);
29
30     @BeforeClass
31     public static void setUpBeforeClass() throws Exception {
32     }
33
34     @AfterClass
35     public static void tearDownAfterClass() throws Exception {
36     }
37
38     @Before
39     public void setUp() throws Exception {
40     }
41
42     @After
43     public void tearDown() throws Exception {
44     }
45
46     @Test
47     public void test01SetPermission() {
48         PdfUtil target = new PdfUtil();
49         File pdfFile = new File("./data/decrypted.pdf");
50         File outDir = new File("./data/SetPermission.pdf");
51         try {
52             PDDocument orgDocument = PDDocument.load(pdfFile);
53             orgDocument = target.setPermission(orgDocument);
54             orgDocument.save(outDir);
55         } catch (IOException | COSVisitorException e) {
56             e.printStackTrace();
57             fail();
58         }
59     }
60
61     @Test
62     public void test02UserPassword() {
63         PdfUtil target = new PdfUtil();
64         File pdfFile = new File("./data/decrypted.pdf");
65         File outDir = new File("./data/SetUserPassword.pdf");
66         try {
67             PDDocument orgDocument = PDDocument.load(pdfFile);
68             orgDocument = target.setUserPassword(orgDocument, "password");
69             orgDocument.save(outDir);
70         } catch (IOException | COSVisitorException e) {
71             e.printStackTrace();
72             fail();
73         }
74     }
75
76     @Test
77     public void test03Rename() {
78         PdfUtil target = new PdfUtil();
79         File in = new File("./data/SetPermission.pdf");
80         File out = new File("./data/output2.pdf");
81         try {
82             PDDocument orgDocument = target.parseDocument(in);
83             orgDocument.save(out);
84         } catch (IOException | COSVisitorException e) {
85             e.printStackTrace();
86             fail();
87         }
88     }
89
90     @Test
91     public void test04ParseDocument() throws IOException {
92         PdfUtil target = new PdfUtil();
93         PDDocument orgDocument = null;
94         try {
95             File pdfFile = new File("./data/decrypted.pdf");
96             orgDocument = target.parseDocument(pdfFile);
97         } catch (IOException e) {
98             e.printStackTrace();
99         }
100     }
101

```

```

102     fail();
103 } finally {
104     if (orgDocument != null) {
105         orgDocument.close();
106     }
107 }
108
109 try {
110     File pdfFile = new File("../data/SetUserPassword.pdf");
111     orgDocument = target.parseDocument(pdfFile);
112 } catch (IOException e) {
113     e.printStackTrace();
114     fail();
115 } finally {
116     if (orgDocument != null) {
117         orgDocument.close();
118     }
119 }
120 }
121
122 @Test
123 public void test05ParseDocument() {
124     PdfUtil target = new PdfUtil();
125     try {
126         File pdfFile = new File("../data/SetPermission.pdf");
127         PDDocument orgDocument = target.parseDocument(pdfFile);
128         orgDocument.save("../data/output2.pdf");
129     } catch (IOException | COSVisitorException e) {
130         e.printStackTrace();
131         fail();
132     }
133 }
134
135
136 @Test
137 public void test06CreatePdfThisPageNumber() {
138     PdfUtil target = new PdfUtil();
139     File pdfFile = new File("../data/encrypted.pdf");
140     File outDir = new File("../data");
141     int rotation = 90;
142     try {
143         //1ページ目を切り出し、右に90度回転
144         target.createPdfThisPage(pdfFile, 1, outDir, rotation);
145     } catch (Throwable e) {
146         e.printStackTrace();
147         fail();
148     }
149 }
150
151 @Test
152 public void test07CreatePdfThisPageNumber2() {
153     PdfUtil target = new PdfUtil();
154     File pdfFile = new File("../data/decrypted.pdf");
155     File outDir = new File("../data");
156     int rotation = 90;
157     try {
158         target.createPdfThisPage(pdfFile, 2, outDir, rotation);
159     } catch (IOException e) {
160         e.printStackTrace();
161         fail();
162     }
163 }
164
165 @Test
166 public void test08CreatePdfThisPageNumber3() {
167     PdfUtil target = new PdfUtil();
168     File pdfFile = new File("../data/decrypted.pdf");
169     File outDir = new File("../data");
170     int rotation = 90;
171     try {
172         target.createPdfThisPage(pdfFile, 3, outDir, rotation);
173     } catch (IOException e) {
174         e.printStackTrace();
175         fail();
176     }
177 }
178
179 @Test
180 public void test09PdfToImage() {
181     PdfUtil target = new PdfUtil();
182     File pdfFile = new File("../data/encrypted.pdf");

```

```
183 File out = new File("./data/encrypted.jpg");
184 try {
185     List<BufferedImage> list = target.pdfToImage(pdfFile, BufferedImage.TYPE_INT_RGB);
186     BufferedImage im = list.get(0);
187     ImageIOUtil.writeImage(im, "jpg", new FileOutputStream(out));
188 } catch (IOException e) {
189     e.printStackTrace();
190     fail();
191 }
192 }
193
194 @Test
195 public void test10MergePdfFile() {
196     PdfUtil target = new PdfUtil();
197     try {
198         List<PdfPageModel> pages = new ArrayList<PdfPageModel>();
199         pages.add(new PdfPageModel(new File("./data/N01.pdf"), 1, new BufferedImage(10, 10,
200 BufferedImage.TYPE_BYTE_GRAY), 0));
201         pages.add(new PdfPageModel(new File("./data/encrypted.pdf"), 1, new BufferedImage(10, 10,
202 BufferedImage.TYPE_BYTE_GRAY), 0));
203         target.mergePdfPageModelToPdfFile(pages, new File("./data/output.pdf"));
204     } catch (IOException e) {
205         e.printStackTrace();
206         fail();
207     }
208 }
209 }
210 }
```