1. PDF Renderer

URL	https://pdf-renderer.dev.java.net/
ライセンス	LGPL
最新バージョン	1.0.5

PDF Renderer は Java で実装された PDF 表示用のライブラリです。Swing アプリケーションで PDF を表示したり、PDF を画像に変換したりすることができます。

ただし、複雑な PDF は正しくレンダリングできない場合もあるほか、日本語フォントのレンダリングに問題があるなど、レンダリングの精度や品質を求められる場面での利用はおすすめできません。利用にあたっては注意してください。

1.1. インストール方法

PDF Renderer の Web サイトのトップページにある[download binary jar]リンクから PDFRenderer.-jar をダウンロードし、クラスパスに追加します。

Maven を利用する場合

Maven から利用する場合、pom.xml に以下の依存関係を追加します。

```
<dependency>
  <groupId>org.swinglabs</groupId>
  <artifactId>pdf-renderer</artifactId>
  <version>1.0.5</version>
</dependency>
```

1.2. PDF ビューアのサンプル

PDF Renderer の JAR ファイルはライブラリとして使用するだけでなく、Java アプリケーションとして実行することで簡易的な PDF ビューアとして使用することができます。PDF ビューアとして起動するには PDFRenderer.jar をダブルクリックするか、コマンドラインから以下のようにして実行します。

```
> java -jar PDFRenderer.jar
```

すると以下のような Swing ベースのアプリケーションが起動します。

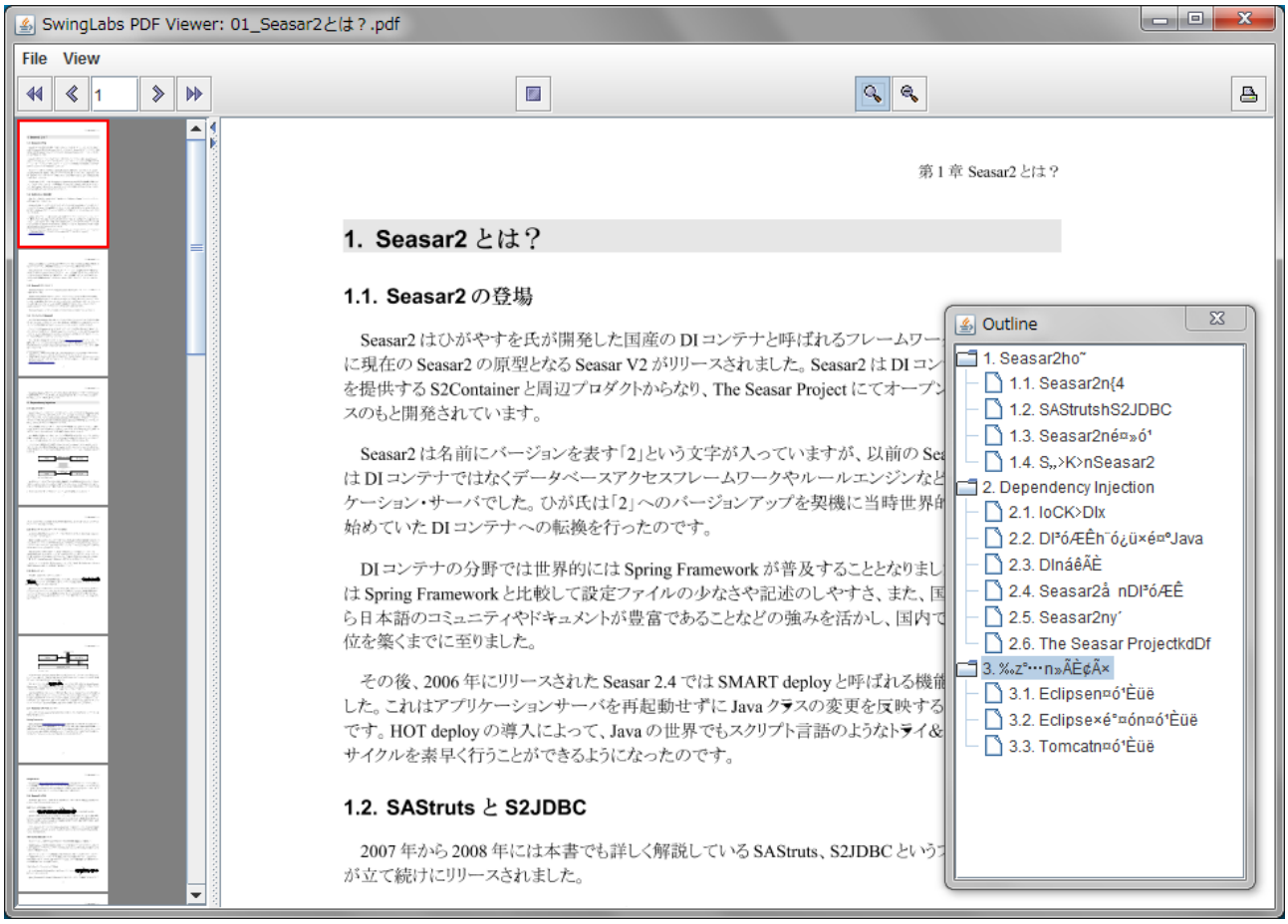


図 1:PDF Renderer に含まれるサンプル PDF ビューア

PDF の見出しマップがポップアップウィンドウで表示されますが、日本語が文字化けしてしまっています。これは PDF Renderer のサンプル PDF ビューアの実装に問題があるようです。API レベルでは見出しの内容を文字化けせずに取得することができます。

1.3. ライブラリとしての PDF Renderer

com.sun.pdfview.PDFFile オブジェクト

PDF Renderer では PDF を com.sun.pdfview.PDFFile オブジェクトで表します。PDFFile オブジェクトは以下のようにして生成します。

```
// PDF ファイルを読み込み
File file = new File("test.pdf");
RandomAccessFile raf = new RandomAccessFile(file, "r");
FileChannel channel = raf.getChannel();
ByteBuffer buf = channel.map(FileChannel.MapMode.READ_ONLY, 0, channel.size());
PDFFile pdfFile = new PDFFile(buf);
```

Swing アプリケーションへの組み込み

前述の通り、PDF Renderer にも PDF ビューアのサンプルが含まれていますが、PDF Renderer を使用して PDF の内容を表示する例として、Swing ベースのよりシンプルな PDF ビューアを実装してみましょう。

```
public class PDFViewer extends JFrame {

  private static final long serialVersionUID = 1L;

  private PDFFile pdfFile;
  private PagePanel pdfPanel;
  private int pageNum = 1;

  public PDFViewer(PDFFile pdfFile) {
    setTitle("PDF ビューア");
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

    this.pdfFile = pdfFile;

    // PDF の内容を表示するためのパネル
    pdfPanel = new PagePanel();
    add(pdfPanel, BorderLayout.CENTER);

    JButton next = new JButton("次のページ");
    next.addActionListener(new ActionListener() {
      @Override
      public void actionPerformed(ActionEvent e) {
        showNextPage();
      }
    });

    JButton prev = new JButton("前のページ");
    prev.addActionListener(new ActionListener() {
      @Override
      public void actionPerformed(ActionEvent e) {
        showPrevPage();
      }
    });

    JPanel bottomBar = new JPanel();
    bottomBar.add(prev);
    bottomBar.add(next);
    add(bottomBar, BorderLayout.SOUTH);

    pack();
    setVisible(true);

    // PDF の 1 ページ目を表示
```

```
    PDFPage firstPage = pdfFile.getPage(1);
    pdfPanel.showPage(firstPage);
  }

  // 次のページを表示
  private void showNextPage() {
    if (pdfFile.getNumPages() > pageNum) {
      pageNum++;
      PDFPage nextPage = pdfFile.getPage(pageNum);
      pdfPanel.showPage(nextPage);
    }
  }

  // 前のページを表示
  private void showPrevPage() {
    if (pageNum > 1) {
      pageNum--;
      PDFPage prevPage = pdfFile.getPage(pageNum);
      pdfPanel.showPage(prevPage);
    }
  }

  // 起動用のメソッド
  public static void main(final String[] args) throws Exception {
    // PDF ファイルを読み込み
    File file = new File("test.pdf");
    RandomAccessFile raf = new RandomAccessFile(file, "r");
    FileChannel channel = raf.getChannel();
    ByteBuffer buf = channel.map(FileChannel.MapMode.READ_ONLY, 0, channel.size());
    final PDFFile pdfFile = new PDFFile(buf);

    SwingUtilities.invokeLater(new Runnable() {
      public void run() {
        // PDF ビューアを表示
        new PDFViewer(pdfFile);
      }
    });
  }

}
```

上記のサンプルでは com.sun.pdfview.PagePanel というコンポーネントを使用して PDFFile#get-Page()メソッドで取得した com.sun.pdfview.PDFPage を表示しています。

このアプリケーションを実行すると以下のようなウィンドウが表示され、PDF の内容を確認することができます。

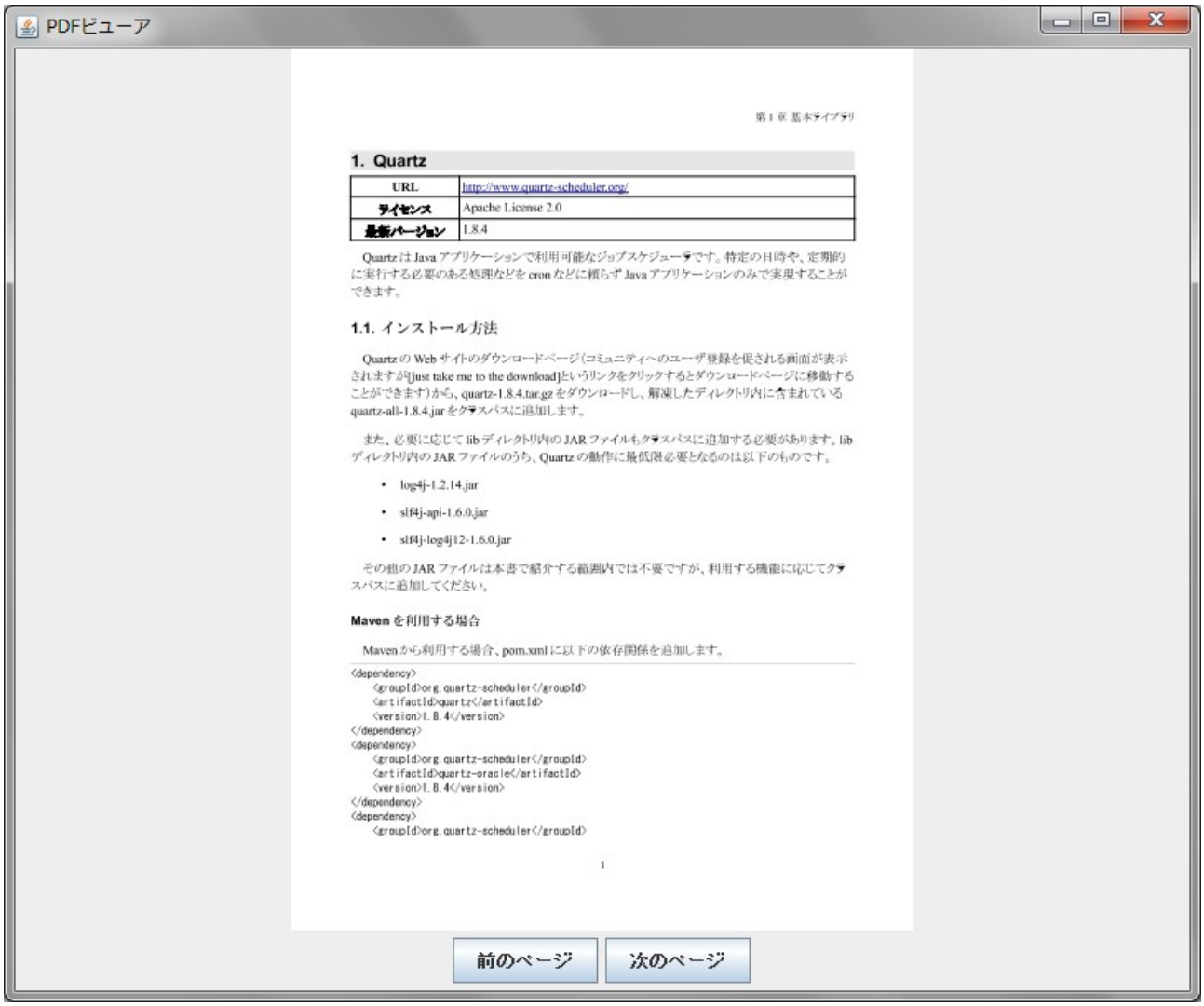


図 2:Swing アプリケーションでの PDF の表示例

PDF を画像に変換する

PDFPage#getImage()メソッドで PDF の該当のページのレンダリング結果を java.awt.Image オブジェクトとして取得することができます。また、com.sun.pdfview.PDFRenderer を使用すると PDFPage の内容を Graphics2D オブジェクトに描画することができます。

以下は PDFRenderer を使用して Graphics2D オブジェクトに描画した PDF のレンダリング結果をファイルとして保存するサンプルです。

```
// PDF ファイルを読み込み
File file = new File("test.pdf");
RandomAccessFile raf = new RandomAccessFile(file, "r");
FileChannel channel = raf.getChannel();
ByteBuffer buf = channel.map(FileChannel.MapMode.READ_ONLY, 0, channel.size());
PDFFile pdfFile = new PDFFile(buf);
```

```
// 1ページ目を取得
PDFPage page = pdfFile.getPage(0);

// 1ページ目の幅と高さを取得
int width  = (int) page.getBBox().getWidth();
int height = (int) page.getBBox().getHeight();

// 1ページ目をBufferedImageにレンダリング
BufferedImage img = new BufferedImage(width, height, BufferedImage.TYPE_INT_ARGB);
Graphics2D g2 = img.createGraphics();
g2.setRenderingHint(RenderingHints.KEY_ANTIALIASING, RenderingHints.VALUE_ANTIALIAS_ON);

PDFRenderer renderer = new PDFRenderer(page, g2, new Rectangle(width, height), null,
Color.WHITE);
page.waitForFinish();
renderer.run();

// レンダリング結果をファイルに出力
ImageIO.write(img, "png", new File("test.png"));
```

このコードを実行すると test.pdf の 1 ページ目の内容が test.png という PNG 形式の画像ファイルとして出力されます。