

HTML to PDF Library 리서치

인턴
이우재, 안상영

목차

1. 오픈소스 라이브러리 - 이우재
2. 유료 서비스 API - 안상영
3. 정리

1. 오픈 소스 라이브러리

- openhtmltopdf

1. 오픈 소스 라이브러리

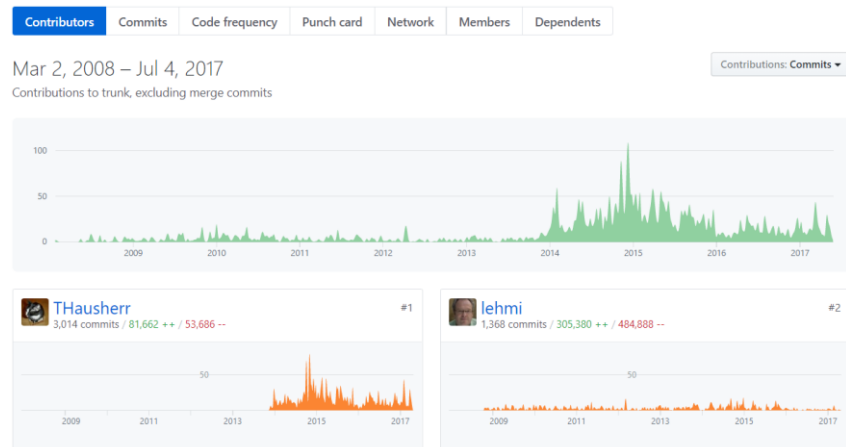
openhtmltopdf - base 1. PDFBox

Base :

- ApachePDFBox(<https://github.com/apache/pdfbox>)
- flyingsaucer(<https://github.com/flyingsaucerproject/flyingsaucer>)

ApachePDFBox :

- pdf document 생성과 변환 가능한 java library
- Apache License v2.0
- 최신 버전 : Apache PDFBox 2.0.6 released (2017-05-15)
- 상세한 tutorial : https://www.tutorialspoint.com/pdfbox/pdfbox_overview.htm
- Include Dependencies Using Maven
- 6261 commits, 6 contributors

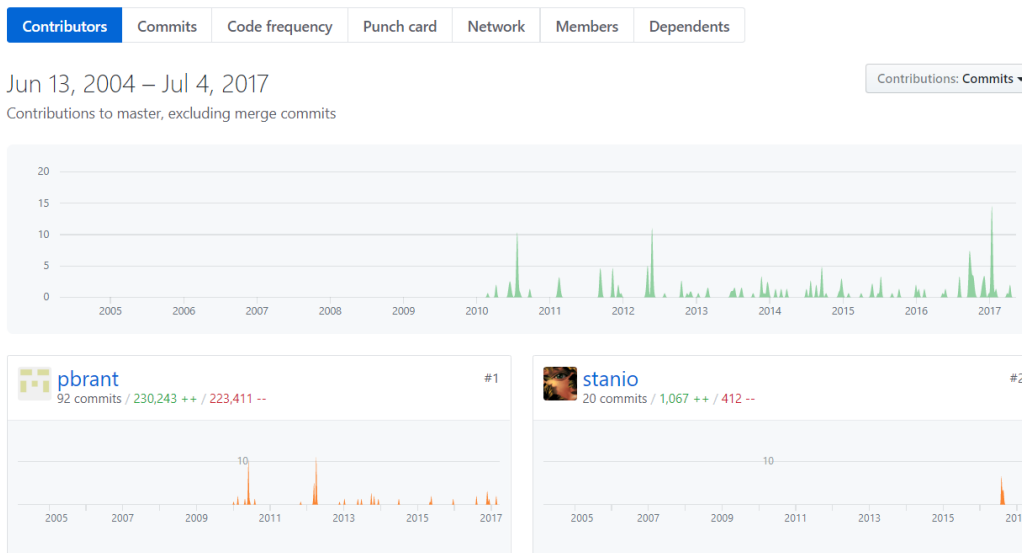


1. 오픈 소스 라이브러리

openhtmltopdf - base 2. flyingsaucer

flyingsaucer :

- XML/XHTML and CSS 2.1 렌더링하는 java library
- GNU Lesser General Public License
- Include Dependencies Using Maven
- 최신버전 : v9.1.5 released(2017-03-29)
- 상세한 tutorial : <https://flyingsaucerproject.github.io/flyingsaucer/r8/guide/users-guide-R8.html>
- 2713 commits, 47 contributors

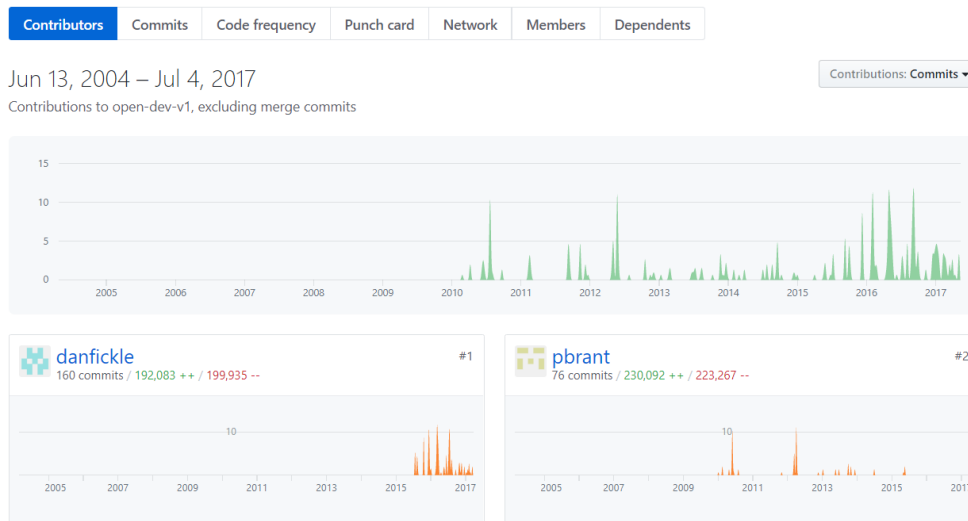


1. 오픈 소스 라이브러리

openhtmltopdf

openhtmltopdf :

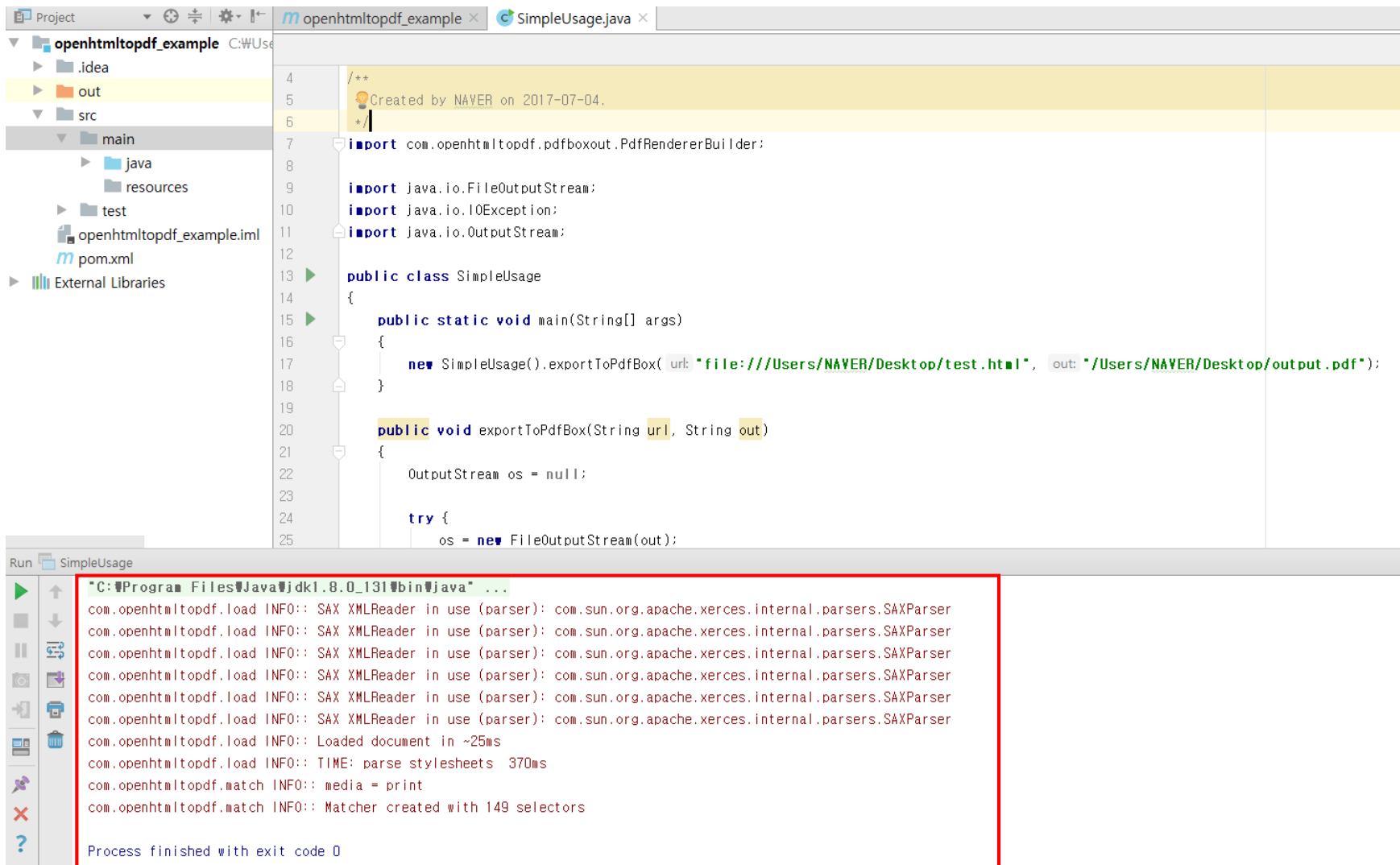
- HTML to PDF library for the JVM
- 최신버전 : 0.0.1-RC11(2017-05-21)
- Include Dependencies Using Maven
- GNU Lesser General Public License, version 2.1 or later
- Guide : <https://github.com/danfickle/openhtmltopdf/blob/open-dev-v1/docs/integration-guide.md>
- 2783 commits, 32 contributors



1. 오픈 소스 라이브러리

openhtmltopdf

Sample code (local html file -> pdf)



The screenshot shows an IDE with a project named 'openhtmltopdf_example'. The 'src/main/java' directory contains a file 'SimpleUsage.java'. The code in this file imports 'com.openhtmltopdf.pdfboxout.PdfRendererBuilder', 'java.io.FileOutputStream', 'java.io.IOException', and 'java.io.OutputStream'. It defines a 'SimpleUsage' class with a 'main' method and an 'exportToPdfBox' method. The 'main' method calls 'exportToPdfBox' with a local HTML file path and an output PDF path. The 'exportToPdfBox' method creates a 'PdfRendererBuilder', sets the 'url' and 'out' paths, and calls 'renderToPdf()'. The execution output at the bottom shows the process running successfully, with various log messages from the 'openhtmltopdf' library and a final message indicating the process finished with exit code 0.

```
4  /**
5  * Created by NAYER on 2017-07-04.
6  */
7  import com.openhtmltopdf.pdfboxout.PdfRendererBuilder;
8
9  import java.io.FileOutputStream;
10 import java.io.IOException;
11 import java.io.OutputStream;
12
13 public class SimpleUsage
14 {
15     public static void main(String[] args)
16     {
17         new SimpleUsage().exportToPdfBox( url: "file:///Users/NAYER/Desktop/test.html", out: "/Users/NAYER/Desktop/output.pdf");
18     }
19
20     public void exportToPdfBox(String url, String out)
21     {
22         OutputStream os = null;
23
24         try {
25             os = new FileOutputStream(out);
```

Run SimpleUsage

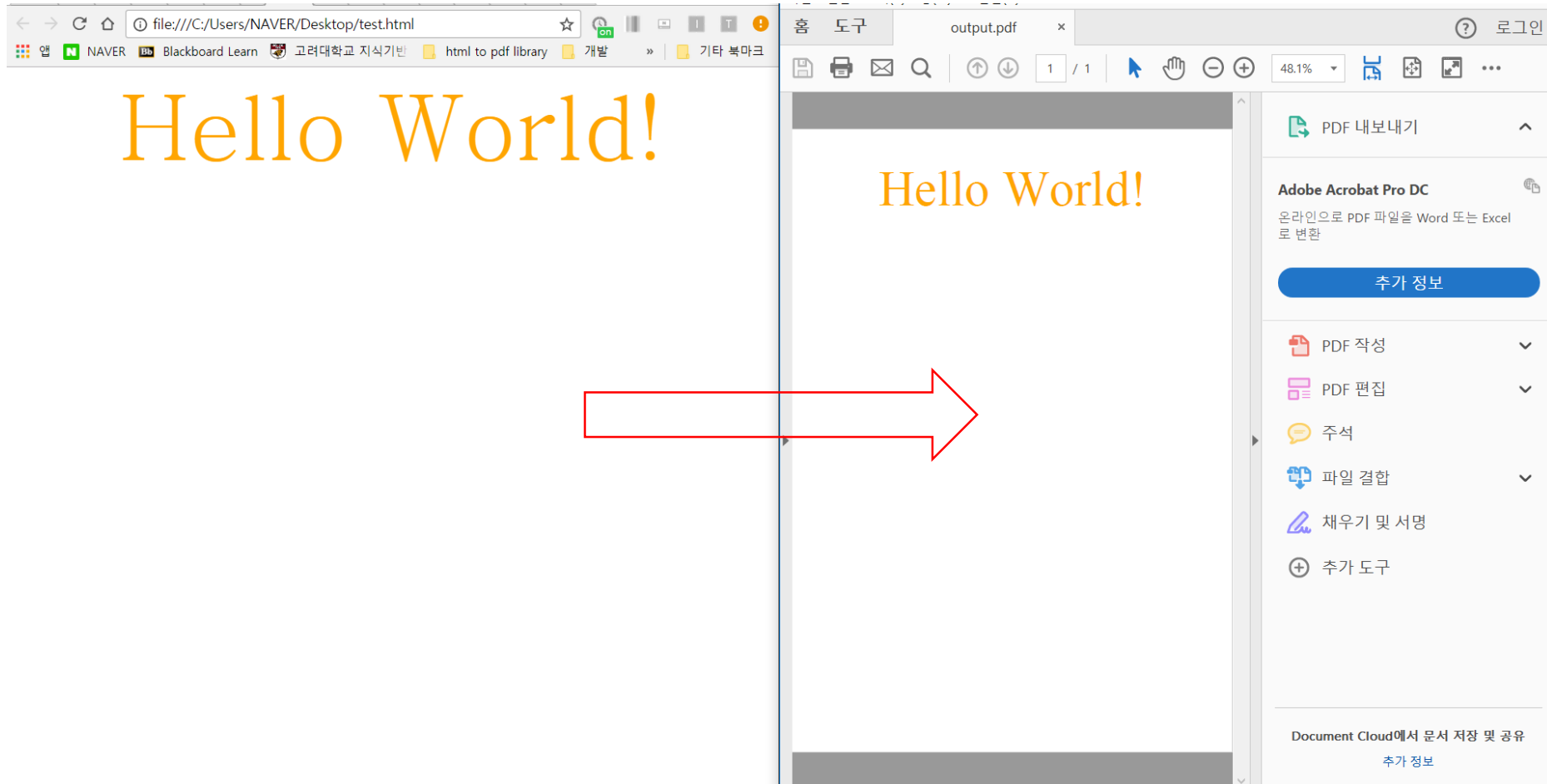
```
"C:\Program Files\Java\jdk1.8.0_131\bin\java" ...
com.openhtmltopdf.load INFO:: SAX XMLReader in use (parser): com.sun.org.apache.xerces.internal.parsers.SAXParser
com.openhtmltopdf.load INFO:: SAX XMLReader in use (parser): com.sun.org.apache.xerces.internal.parsers.SAXParser
com.openhtmltopdf.load INFO:: SAX XMLReader in use (parser): com.sun.org.apache.xerces.internal.parsers.SAXParser
com.openhtmltopdf.load INFO:: SAX XMLReader in use (parser): com.sun.org.apache.xerces.internal.parsers.SAXParser
com.openhtmltopdf.load INFO:: SAX XMLReader in use (parser): com.sun.org.apache.xerces.internal.parsers.SAXParser
com.openhtmltopdf.load INFO:: SAX XMLReader in use (parser): com.sun.org.apache.xerces.internal.parsers.SAXParser
com.openhtmltopdf.load INFO:: Loaded document in ~25ms
com.openhtmltopdf.load INFO:: TIME: parse stylesheets 370ms
com.openhtmltopdf.match INFO:: media = print
com.openhtmltopdf.match INFO:: Matcher created with 149 selectors

Process finished with exit code 0
```

1. 오픈 소스 라이브러리

openhtmltopdf

Sample output



openhtmltopdf

확장하기

PDFBox library

- Adding Pages
- Adding Text
- Inserting Image
- ..

| PDFBox Tutorial |
|-----------------------------------|
| PDFBox - Home |
| PDFBox - Overview |
| PDFBox - Environment |
| PDFBox - Creating a PDF Document |
| PDFBox - Adding Pages |
| PDFBox - Loading a Document |
| PDFBox - Removing Pages |
| PDFBox - Document Properties |
| PDFBox - Adding Text |
| PDFBox - Adding Multiple Lines |
| PDFBox - Reading Text |
| PDFBox - Inserting Image |
| Encrypting a PDF Document |
| JavaScript in PDF Document |
| PDFBox - Splitting a PDF Document |
| Merging Multiple PDF Documents |
| PDFBox - Extracting Image |
| PDFBox - Adding Rectangles |

In the previous chapter, we have seen how to extract text from an existing PDF document. In this chapter, we will discuss how to insert image to a PDF document.

Inserting Image to a PDF Document

You can insert an image into a PDF document using the **createFromFile()** and **drawImage()** methods of the classes **PDIImageXObject** and **PDPageContentStream** respectively.

Following are the steps to extract text from an existing PDF document.

Step 1: Loading an Existing PDF Document

Load an existing PDF document using the static method **load()** of the **PDDocument** class. This method accepts a file object as a parameter, since this is a static method you can invoke it using class name as shown below.

```
File file = new File("path of the document")
PDDocument doc = PDDocument.load(file);
```

Step 2: Retrieving a Page

Select a page in the PDF document and retrieve its page object using the **getPage()** method as shown below.

```
PDPage page = doc.getPage(0);
```

Step 3: Creating PDIImageXObject object

The class **PDIImageXObject** in PDFBox library represents an image. It

1. 오픈 소스 라이브러리

openhtmltopdf

해결해야 할 이슈

xhtml이 아닌 html 형식 convert시 error 발생

- xhtml : 속성 값의 경우 ""를 써줘야 함, 종료 태그 필수

-

"C:\Program Files\Java\jdk1.8.0_131\bin\java" ...

```
com.openhtmltopdf.load INFO:: SAX XMLReader in use (parser): com.sun.org.apache.xerces.internal.parsers.SAXParser
com.openhtmltopdf.load INFO:: SAX XMLReader in use (parser): com.sun.org.apache.xerces.internal.parsers.SAXParser
com.openhtmltopdf.load INFO:: SAX XMLReader in use (parser): com.sun.org.apache.xerces.internal.parsers.SAXParser
com.openhtmltopdf.load INFO:: SAX XMLReader in use (parser): com.sun.org.apache.xerces.internal.parsers.SAXParser
com.openhtmltopdf.load INFO:: SAX XMLReader in use (parser): com.sun.org.apache.xerces.internal.parsers.SAXParser
com.openhtmltopdf.load INFO:: SAX XMLReader in use (parser): com.sun.org.apache.xerces.internal.parsers.SAXParser
com.openhtmltopdf.load INFO:: 요소 유형 "link"은(는) 짝이 맞는 종료 태그 "</link>"(으)로 종료되어야 합니다.
ERROR: '요소 유형 "link"은(는) 짝이 맞는 종료 태그 "</link>"(으)로 종료되어야 합니다.'
com.openhtmltopdf.exception WARNING:: Unhandled exception. Can't load the XML resource (using TRaX transformer). org.xml
com.openhtmltopdf.util.XRRuntimeException: Can't load the XML resource (using TRaX transformer). org.xml.sax.SAXParseException
at com.openhtmltopdf.resource.XMLResource$XMLResourceBuilder.createXMLResource(XMLResource.java:235)
at com.openhtmltopdf.resource.XMLResource$XMLResourceBuilder.access$100(XMLResource.java:164)
at com.openhtmltopdf.resource.XMLResource.load(XMLResource.java:81)
at com.openhtmltopdf.swing.NaiveUserAgent.getXMLResource(NaiveUserAgent.java:368)
at com.openhtmltopdf.pdfboxout.PdfBoxRenderer.loadDocument(PdfBoxRenderer.java:323)
at com.openhtmltopdf.pdfboxout.PdfBoxRenderer.setDocumentP(PdfBoxRenderer.java:335)
```

HTML5 PARSER SUPPORT

While Open HTML to PDF works with a standard w3c DOM, the project provides a converter from the Jsoup HTML5 parser provided Document to a w3c DOM Document. This allows you to parse and use HTML5, rather than the default strict XML required by the project.

Then you can use one of the `Jsoup.parse` methods to parse HTML5 and `DOMBuilder.jsoup2DOM` to convert the Jsoup document to a w3c DOM one.

```
public org.w3c.dom.Document html5ParseDocument(String urlStr, int timeoutMs) throws IOException
{
    URL url = new URL(urlStr);
    org.jsoup.nodes.Document doc;

    if (url.getProtocol().equalsIgnoreCase("file")) {
        doc = Jsoup.parse(new File(url.getPath()), "UTF-8");
    }
    else {
        doc = Jsoup.parse(url, timeoutMs);
    }

    return DOMBuilder.jsoup2DOM(doc);
}
```

Then you can set the renderer document with `builder.withW3cDocument(doc, url)` in place of `builder.withUri(url)`.

2. 유료 서비스 API

(1) DocRaptor

(2) ITEXT

(3) GRAZIT

2. 유료 서비스 API

1-1. DocRaptor - 소개

- The best API for creating PDF and XLS files
- HTML, JS, CSS를 전송 => PDF

The screenshot shows the DocRaptor website homepage. At the top, there's a navigation bar with the DocRaptor logo (a blue dinosaur) and links for 'How It Works', 'Samples', 'Try It Out!', 'Login', and 'Pricing & Free Trial'. The main hero section has a blue background with the text 'The best API for creating PDF and XLS files' and a subtext 'You just send HTML, JS, and CSS, and you get back beautiful, high quality documents at scale.' Below this, there are three bullet points: 'Uptime guaranteed', 'Unlimited document size', and 'Expert support, including document debugging'. To the right, there's a diagram showing an HTML document being converted into XLS and PDF files. At the bottom of the hero section, there are two buttons: 'Start your Free Trial' and 'How It Works'. Below the hero section, there are logos for 'shopify', 'accenture', 'HubSpot', and 'Keen IO'. A quote below these logos says 'Our HTML to PDF API is used by companies big and small'. At the very bottom, there's a visual representation of a conveyor belt with several document icons (HTML, XLS, PDF) moving along it, with the DocRaptor logo in the center.

DocRaptor

How It Works Samples Try It Out! Login Pricing & Free Trial

The best API for creating PDF and XLS files

You just send HTML, JS, and CSS, and you get back beautiful, high quality documents at scale.

- Uptime guaranteed
- Unlimited document size
- Expert support, including document debugging

Start your Free Trial How It Works

shopify accenture HubSpot Keen IO

Our HTML to PDF API is used by companies big and small

1-2. DocRaptor - 특징

- 높은 인지도 (개발자 커뮤니티에서 자주 언급)
- POST request 수준의 쉬운 인터페이스
- 업타임 보증
- 무한대의 도큐먼트 사이즈
- 커스텀 폰트 & 타이포그래피
- Advanced 레이아웃 (headers, footers, page numbers, tables, lists, columns, floats, footnotes, cross-reference)
- 도큐먼트 디버깅을 포함한 전문가 서포트
- Prince (HTML 조작) Engine으로 렌더링 => 문서 포맷

1-3. DocRaptor - API overview

```
// Add to pom.xml
// <dependency>
//   <groupId>com.docraptor</groupId>
//   <artifactId>docraptor</artifactId>
//   <version>1.1.0</version>
// </dependency>
//
// or get JARs from https://github.com/DocRaptor/docraptor-java/releases
import java.io.*;
import java.net.*;
import com.docraptor.*;

public class Sync {
    public static void main(String[] args) throws Exception {
        DocApi docraptor = new DocApi();
        ApiClient client = docraptor.getApiClient();
        client.setUsername("YOUR_API_KEY_HERE");

        Doc doc = new Doc();
        doc.setTest(true);
        doc.setDocumentUrl("http://www.docraptor.com/examples/invoice.html");
        doc.setDocumentType(Doc.DocumentTypeEnum.PDF);
        doc.setName("docraptor-java.pdf");

        docraptor.createDoc(doc);
    }
}
```

1-4. DocRaptor - Documentation

Documentation

- </> Code Examples
- 📄 Style & Formatting
- i API Reference**

- Document API**
 - Authentication
 - Making a Document
 - API Parameters
 - HTTP Status Codes
 - Response Headers
 - Limits
 - Test Documents
 - Support & Debugging
- Referrer-Based Documents**
- Document Listing API**
- IP Listing API**

Document API

Converting HTML into PDF and XLS documents is fast and painless with DocRaptor. Browse the documentation below to get started, or check out a [code example in your language](#).

Authentication

You'll authenticate with your API key, which can be found on your [account dashboard](#). You can use your API key in one of two ways:

- The username for HTTP Basic Authentication (preferred)
- The value for the query parameter `user_credentials`

If you're using our [official clients](#), authentication is included in all examples.

Making a Document

To make documents, **POST** to <https://docraptor.com/docs> with **JSON**. Only a few options are required and most have reasonable defaults. Here are the minimum options for making a PDF:

```
{
  "type": "pdf",
  "document_content": "<html><body>Hello World!</body></html>"
}
```

Some options are nested and are referred to like `prince[media]` in the documentation. As **JSON** they should look like this:

1-4. DocRaptor - Pricing

골드 플랜 기준:
2.25 ¢ per doc
약 23원

| PLAN | DOCUMENTS | PRICE |
|---|---------------------------|--------------------------|
| <input checked="" type="radio"/> Basic | 125 <small>mo</small> | \$15 <small>mo</small> |
| <input type="radio"/> Professional | 325 <small>mo</small> | \$29 <small>mo</small> |
| <input type="radio"/> Premium | 1,250 <small>mo</small> | \$75 <small>mo</small> |
| <input type="radio"/> Max | 5,000 <small>mo</small> | \$149 <small>mo</small> |
| <input type="radio"/> Bronze | 15,000 <small>mo</small> | \$399 <small>mo</small> |
| <input type="radio"/> Silver | 40,000 <small>mo</small> | \$1000 <small>mo</small> |
| <input type="radio"/> Gold | 100,000 <small>mo</small> | \$2250 <small>mo</small> |
| <input type="radio"/> Enterprise | ∞ | Contact us! |

Overage price for the Basic Plan: 12¢ per doc

2. 유료 서비스 API

2-1. ITEXT - 소개

- PDF 기능들을 app, process, produc에 통합시킬 수 있는 소프트웨어 개발 툴킷
- 앞서 소개한 pdfbox의 유료 서비스판으로 이해 가능

[Blog & PR](#) | [Support](#) | [About](#) | [Contact](#) | [English](#) ▼

Why iText? +

[Products & Resources »](#)[Developers »](#)[Pricing »](#)[Request a quote »](#)[Free Trial »](#)

2. 유료 서비스 API

2-2. ITEXT - 특징

- 높은 인지도 (개발자 커뮤니티에서 자주 언급)
 - Html to pdf 포팅뿐만 아니라 PDF 관련 기능들을 모두 구현 가능한 강력한 기능
 - 방대한 개발 문서
 - 오래되었지만 (15년) 지속적인 업데이트 및 서비스
 - 수동적인 document 프로세스 제거
 - Workflow 퍼포먼스 향상
 - 산업 및 법적 표준 준수
 - 4가지 비용 산출 방식
- iText 7 Converting HTML to PDF
 - pdfHTML: Accessible PDF Creation
 - pdfHTML: Colourblindness example
 - pdfHTML: Custom Tagworker example
 - pdfHTML: Header and Footer example
 - pdfHTML: QRCode Example
 - pdfHTML: Responsive Design

2-3. ITEXT - API Overview

```
/*
   This file is part of the iText (R) project.
   Copyright (c) 1998-2017 iText Group NV
   Authors: iText Software.

   For more information, please contact iText Software at this address:
   sales@itextpdf.com
 */
package com.itextpdf.samples.pdfHTML.AccessiblePDF;

import com.itextpdf.html2pdf.ConverterProperties;
import com.itextpdf.html2pdf.HtmlConverter;
import com.itextpdf.html2pdf.attach.impl.DefaultTagWorkerFactory;
import com.itextpdf.kernel.pdf.PdfDocument;
import com.itextpdf.kernel.pdf.PdfDocumentInfo;
import com.itextpdf.kernel.pdf.PdfName;
import com.itextpdf.kernel.pdf.PdfString;
import com.itextpdf.kernel.pdf.PdfViewerPreferences;
import com.itextpdf.kernel.pdf.PdfWriter;
import com.itextpdf.kernel.pdf.WriterProperties;
import com.itextpdf.layout.font.FontProvider;
import com.itextpdf.licensekey.LicenseKey;
import com.itextpdf.samples.pdfHTML.AccessiblePDF.HeaderTagging.AccessibilityTagWorkerFactory;

import java.io.File;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;

public class CreateAccessiblePDF {

    public static final String sourceFolder = "src/test/resources/pdfHTML/AccessiblePDF/";
    public static final String destinationFolder = "target/output/pdfHTML/AccessiblePDF/";
    public static final String[] files = {"Accessibility"};
    //License key path
    public static final String LICENSE = "src/test/resources/pdfHTML/itextkey-html2pdf_typography.xml";
```

2-3. ITEXT - API Overview

```
public void createPdf(String src, String dest, String resources) throws IOException {
    try {
        FileOutputStream outputStream = new FileOutputStream(dest);

        WriterProperties writerProperties = new WriterProperties();
        //Add metadata
        writerProperties.addXmpMetadata();

        PdfWriter pdfWriter = new PdfWriter(outputStream, writerProperties);

        PdfDocument pdfDoc = new PdfDocument(pdfWriter);
        pdfDoc.getCatalog().setLang(new PdfString("en-US"));
        //Set the document to be tagged
        pdfDoc.setTagged();
        pdfDoc.getCatalog().setViewerPreferences(new PdfViewerPreferences().setDisplayDocTitle(true)
    );

    //Set meta tags
    PdfDocumentInfo pdfMetaData = pdfDoc.getDocumentInfo();
    pdfMetaData.setAuthor("Samuel Huylebroeck");
    pdfMetaData.addCreationDate();
    pdfMetaData.getProducer();
    pdfMetaData.setCreator("iText Software");
    pdfMetaData.setKeywords("example, accessibility");
    pdfMetaData.setSubject("PDF accessibility");
    //Title is derived from html

    // pdf conversion
    ConverterProperties props = new ConverterProperties();
    FontProvider fp = new FontProvider();
    fp.addStandardPdfFonts();
    fp.addDirectory(resources); //The noto-nashk font file (.ttf extension) is placed in the resources

    props.setFontProvider(fp);
    props.setBaseUri(resources);
    //Setup custom tagworker factory for better tagging of headers
    DefaultTagWorkerFactory tagWorkerFactory = new AccessibilityTagWorkerFactory();
    props.setTagWorkerFactory(tagWorkerFactory);

    HtmlConverter.convertToPdf(new FileInputStream(src), pdfDoc, props);
    pdfDoc.close();
}
```

2-3. ITEXT - API Overview

| File name | Raw URL | Updated |
|--|--|---------------------|
| CreateAccessiblePDF.java | CreateAccessiblePDF.java | 2017-06-07 10:43 am |
| AccessibilityTagWorkerFactory.java | AccessibilityTagWorkerFactory.java | 2017-06-07 10:54 am |
| HeaderTagWorker.java | HeaderTagWorker.java | 2017-06-07 10:54 am |
| TableHeaderTagWorker.java | TableHeaderTagWorker.java | 2017-06-07 10:54 am |

2-4. ITEXT - 개발 문서

ITEXT Developers

Blog & PR | Support | Pricing | Contact

English

Search

REFERENCES

EXAMPLES

Q&A

TUTORIALS

DOWNLOADS

MAIN SITE

iText 7: Jump-Start Tutorial

Before we start: installing iText 7

Chapter 1: Introducing basic building blocks

Chapter 2: Adding low-level content

Chapter 3: Using renderers and event handlers

Chapter 4: Making a PDF interactive

Chapter 5: Manipulating an existing PDF document

Chapter 6: Reusing existing PDF documents

Chapter 7: Creating PDF/UA and PDF/A documents

Examples

Chapter 1

Chapter 2

Chapter 3

Chapter 4

Chapter 5

iText 7: Jump-Start Tutorial / Examples / Chapter 1

Chapter 1

Files:

C01E01_HelloWorld.java

```
/*
 * This example is part of the iText 7 tutorial.
 */
package tutorial.chapter01;

import com.itextpdf.kernel.pdf.PdfDocument;
import com.itextpdf.kernel.pdf.PdfWriter;
import com.itextpdf.layout.Document;
import com.itextpdf.layout.element.Paragraph;
import com.itextpdf.test.annotations.WrapToTest;

import java.io.File;
import java.io.IOException;

/**
 * Simple Hello World example.
 */
@WrapToTest
public class C01E01_HelloWorld {

    public static final String DEST = "results/chapter01/hello_world.pdf";

    public static void main(String args[]) throws IOException {
        File file = new File(DEST);
    }
}
```


2-4. ITEXT - 개발 문서

All Classes

Packages

com.itextpdf.barcodes
com.itextpdf.barcodes.dmcode
com.itextpdf.barcodes.qrcode
com.itextpdf.forms
com.itextpdf.forms.fields
com.itextpdf.forms.xfa
com.itextpdf.io
com.itextpdf.io.codec
com.itextpdf.io.color
com.itextpdf.io.font

All Classes

AbstractCMap
AbstractElement
AbstractRenderer
AbstractWidthHandler
AccessibilityProperties
AccessibleAttributesApplier
ActualTextIterator
AdobeGlyphList
AESCipher
AESCipherCBCNoPad
AesDecryptor
AffineTransform
AliasOptions
AlphabetNumbering
ARCF4Encryption
AreaBreak
AreaBreakRenderer
AreaBreakType
ArmenianNumbering
ArrayUtil
ASCII85DecodeFilter
ASCIIHexDecodeFilter
Background
BackgroundImage
BadPasswordException
Barcode128
Barcode128.Barcode128CodeSet
Barcode1D
Barcode2D
Barcode39

OVERVIEW PACKAGE CLASS USE TREE DEPRECATED INDEX HELP

PREV NEXT FRAMES NO FRAMES

iText 7 7.0.3 API

Barcodes

| Package | Description |
|------------------------------|-------------|
| com.itextpdf.barcodes | |
| com.itextpdf.barcodes.dmcode | |
| com.itextpdf.barcodes.qrcode | |

Forms

| Package | Description |
|---------------------------|-------------|
| com.itextpdf.forms | |
| com.itextpdf.forms.fields | |
| com.itextpdf.forms.xfa | |

Input / Output

| Package | Description |
|--------------------------------------|-------------|
| com.itextpdf.io | |
| com.itextpdf.io.codec | |
| com.itextpdf.io.color | |
| com.itextpdf.io.font | |
| com.itextpdf.io.font.cmap | |
| com.itextpdf.io.font.otf | |
| com.itextpdf.io.font.otf.lookuptype5 | |
| com.itextpdf.io.font.otf.lookuptype6 | |
| com.itextpdf.io.image | |

23

2-5. ITEXT - 릴리즈 노트

Release Note and Changelogs

iText 7 Core Changelog

Posted: Tags: [Changelog](#) [iText 7](#)
06/23/2017
June 23, 2017
[Read more](#)

Release notes for 7.0.3

Posted: Tags: [Release iText 7](#)
06/23/2017
With [7.0.3](#) we're bringing you another maintenance release for iText 7 Community. We've also updated pdfSweep to [1.0.2](#) , pdfXFA to [1.0.1](#) and the license key library to [2.0.4](#) . As these are maintenance releases, there are no major new features. Hereby,...
[Read more](#)

iText 7.0.2 release

Posted: Tags: [Release iText 7](#)
02/17/2017
iText 7 Release Core 7.0.2 Available on [NuGet \(.NET\)](#) and on [Maven Central \(Java\)](#) . We have released a minor update to our iText 7 platform, version 7.0.2. In this maintenance release, we will support some upcoming product launches (stay tuned!) as well as bugfixes and performance improvements. Beyond these updates, we have also improved: font support with FontProvider/FontSelector advanced page range definition, contributed by GitHub user @...
[Read more](#)

Categories

- [General](#)
- [Press and news](#)
- [Events](#)
- [Insights and thoughts](#)
- [Release Note and Changelogs](#)

Archive

- [2017 \(41\)](#)
- [2016 \(46\)](#)
- [2015 \(56\)](#)
- [2014 \(62\)](#)
- [2013 \(66\)](#)
- [2012 \(17\)](#)
- [2011 \(12\)](#)
- [2010 \(6\)](#)
- [2009 \(4\)](#)
- [2008 \(6\)](#)
- [2007 \(8\)](#)
- [2006 \(2\)](#)
- [2005 \(1\)](#)
- [2004 \(2\)](#)
- [2003 \(1\)](#)
- [2001 \(1\)](#)
- [2000 \(1\)](#)

2-6. ITEXT - Pricing

Our licensing models



Unit-based licenses

Did you know our licenses are perpetual? By counting the number of servers, clients or devices on which you run iText solutions, you can calculate the price for our most traditional licensing type. Please check it out and request your quote today.



Volume-based licenses

This is the license you need when you can't – or won't – count the number of servers you are using. In this case, we can offer you a license for a yearly fee that is based on the number of documents you are processing with your application.



OEM licenses

This license type allows you to embed iText into your own products. Your customers only use iText functionalities made available through your application and our license doesn't allow them to use iText outside the context of your product.



Maintenance and support

Unit-based and OEM licenses are perpetual and require the additional purchase of first year support at 20% of list price. Support renewals are available at 20% of prevailing list price - 95% of our customers renew their support annually, a testament to our level of care and attention. Volume based licenses are annual subscriptions and include support.

Unit-based license 기준:
서버당 \$3,360
약 386만원

3-1. GRAZBIT - 소개

- Screenshot a website를 슬로건으로 html to pdf conversion 기능 제공

GRABZIT

Search for...

Tools to Capture and Convert the Web

HOME GET STARTED API WEB SCRAPER SCREENSHOT TOOL UPGRADE SUPPORT SIGN IN

SIGN UP

ABOUT US

AFFILIATE PROGRAM

BLOG

CONTACT US

PLUGINS

SERVICE STATUS

Try all our premium features for free with a 7 day free trial. Then from KR\$5.99 a month, unless cancelled.

More Captures

More Features

More API's

Bigger Scrapes

Bigger Screenshots

Bigger Everything

Start Free Trial

Screenshot a Website

URL to Capture e.g. <http://v>

FAST

To ensure your app is fast our entire service is highly optimized for speed, with most screenshots generated in a few seconds. While our [web scraper](#)

FLEXIBLE

To allow you to achieve your goals our service is highly customizable. Create website screenshots ranging from [website thumbnails](#) to [full page](#)

SIMPLE

We want you to get started as quickly as possible, which is why we have simplified our services. You can easily enable your desktop or web app to

RELIABLE

We understand that you are relying on us, which is why GrabzIt has been built from the bottom up with reliability and uptime in mind. with every part of our

ENHANCE YOUR APP OR BUSINESS WITH GRABZIT

Screenshot a Website or Convert HTML into Images - using our API and online Screenshot Tool.

Convert Webpages and URL's to PDF or Convert HTML into PDF's - using our API and online Screenshot Tool.

HTML to DOCX - convert HTML and web pages to DOCX using our API and online Screenshot Tool.

Video to Animated GIF - convert YouTube, Vimeo and online videos into animated GIF's using our API and Screenshot Tool.

HTML Table to CSV - our API and Screenshot Tool can be used to convert HTML tables to CSV or Excel spreadsheets.

Webpage to Icon - generate an image icon for any URL based on a pages metadata with our Webpage to Icon API.

Scrape a Website - use our web scraper to extract data directly from any website and return it in a usable form.

26

3-2. GRAZBIT - 특징

- 간편한 API
- 빈약한 개발 문서
- 저렴한 비용
- 월간 한계 개수 존재 (비용 추가를 통해 한계를 늘릴 수 있는지 문의 필요)
- 블로그를 통한 지속적인 소통 및 릴리즈

3-3. GRAZBIT - API Overview

```
//Create the GrabzItClient class
//Replace "APPLICATION KEY", "APPLICATION SECRET" with the values from your account!
GrabzItClient grabzIt = new GrabzItClient("Sign in to view your Application Key", "Sign in to
view your Application Secret");

// To convert HTML into a image
grabzIt.HTMLToImage("<html><body><h1>Hello World!</h1></body></html>");

// Or convert HTML into a PDF document
grabzIt.HTMLToPDF("<html><body><h1>Hello World!</h1></body></html>");

// Or convert HTML into a CSV, XLSX or JSON document
grabzIt.HTMLToTable("<html><body><table><tr><th>Name</th><th>Age</th></tr><tr><td>Tom</td>
<td>23</td></tr><tr><td>Nicola</td><td>26</td></tr></table></body></html>");

grabzIt.Save("http://www.example.com/handler");
|
String filepath = "images/result.jpg";

grabzIt.SaveTo(filepath);
```


2. 유료 서비스 API

3-4. GRAZBIT - 개발 문서

PDFOptions

The class represents all of the options available when creating PDF captures.

PUBLIC METHODS

- ◆ **setCustomId(String value)** - a custom identifier that you can pass through to the webservice. This will be returned with the callback URL you have specified.
 - ◆ Default: empty
- ◆ **setIncludeBackground(boolean value)** - if true the background of the web page should be included in the screenshot
 - ◆ Default: true
- ◆ **setPageSize(PageSize value)** - the page size of the PDF
 - ◆ Default: `PageSize.A4`
 - ◆ Options: `PageSize.A3`, `PageSize.A4`, `PageSize.A5`, `PageSize.A6`, `PageSize.B3`, `PageSize.B4`, `PageSize.B5`, `PageSize.B6`, `PageSize.LEGAL`, `PageSize.LETTER`
- ◆ **setOrientation(PageOrientation value)** - the orientation of the PDF document
 - ◆ Default: `PageOrientation.PORTRAIT`
 - ◆ Options: `PageOrientation.PORTRAIT`, `PageOrientation.LANDSCAPE`
- ◆ **setIncludeLinks(boolean value)** - true if links should be included in the PDF
 - ◆ Default: true
- ◆ **setIncludeOutline(boolean value)** - true if the PDF outline should be included
 - ◆ Default: false
- ◆ **setTitle(String value)** - provide a title to the PDF document
 - ◆ Default: empty
- ◆ **setCoverURL(string value)** - the URL of a web page that should be used as a cover page for the PDF
 - ◆ Default: empty
- ◆ **setMarginTop(int value)** - the margin in millimeters that should appear at the top of the PDF document page
 - ◆ Default: 10
- ◆ **setMarginLeft(int value)** - the margin in millimeters that should appear at the left of the PDF document page
 - ◆ Default: 10
- ◆ **setMarginBottom(int value)** - the margin in millimeters that should appear at the bottom of the PDF document page
 - ◆ Default: 10
- ◆ **setMarginRight(int value)** - the margin in millimeters that should appear at the right of the PDF document
 - ◆ Default: 10
- ◆ **setDelay(int value)** - the number of milliseconds to wait before taking the screenshot
 - ◆ Default: 0
 - ◆ Maximum: 30000
- ◆ **setRequestAs(BrowserType value)** - the type of user agent you wish to use
 - ◆ Default: `BrowserType.STANDARDROWSER`
 - ◆ Options: `BrowserType.STANDARDROWSER`, `BrowserType.MOBILEBROWSER`, `BrowserType.SEARCHENGINE`, `BrowserType.FALLBACKBROWSER`
- ◆ **setTemplateId(String value)** - add a PDF template ID that specifies the header and footer of the PDF document
 - ◆ Default: empty
- ◆ **setTargetElement(String value)** - the id or class CSS selector of the only HTML element on the target web page that is to be turned into a screenshot, all other parts of the web page are ignored. If there are multiple matching HTML elements the first one is chosen
 - ◆ Default: empty
- ◆ **setHideElement(String value)** - the id or class CSS selectors of the one or more HTML elements in the web page to hide, to specify multiple HTML elements to hide separate each selector with a comma
 - ◆ Default: empty
- ◆ **setWaitForElement(String value)** - the id or class CSS selectors of the HTML element in the web page that must be visible before the capture is performed
 - ◆ Default: empty
- ◆ **setCustomWaterMarkId(String value)** - add a custom watermark to each page of the PDF document
 - ◆ Default: empty
- ◆ **setQuality(int value)** - the quality of the returned PDF. The default uses the recommended quality for the PDF.
 - ▽ Reducing the quality will reduce the filesize and reduce download times.
 - ◆ Default: -1
 - ◆ Minimum: -1
 - ◆ Maximum: 100
- ◆ **setCountry(Country value)** - the country the screenshot should be taken from.
 - ⚠ Warning this feature is currently in beta and may not provide consistent results.
 - ◆ Default: The current fastest location
 - ◆ Options: `Country.SINGAPORE`, `Country.UNITEDKINGDOM`, `Country.UNITEDSTATES`
- ◆ **setExportURL(String value)** - the export URL that specifies where the capture should be exported too
 - ◆ Default: empty

3-5. GRAZBIT - Pricing

| | Free | Entry | Professional | Business | Enterprise |
|-----------------------------|----------------|-------------|--------------|--------------|--------------|
| Standard Features | | | BEST VALUE! | | |
| Maximum Captures Per Month | 1,000 | 5,000 | 50,000 | 250,000 | 500,000 |
| Maximum Requests Per Minute | 60 | 120 | 120 | 150 | 180 |
| Capture Cache Time | Thirty Minutes | Three Hours | Six Hours | Twelve Hours | Twelve Hours |
| Support | Basic | Priority | Priority | Priority | Priority |
| Capture Speed | | | | | |
| High Quality Captures | - | ✓ | ✓ | ✓ | ✓ |
| Customize Delay | - | ✓ | ✓ | ✓ | ✓ |
| Customize Cookies | - | ✓ | ✓ | ✓ | ✓ |
| Customize Watermarks | - | ✓ | ✓ | ✓ | ✓ |
| GrabIt Watermark Removed | - | ✓ | ✓ | ✓ | ✓ |
| All Output Formats | - | ✓ | ✓ | ✓ | ✓ |

Enterprise 맥시멈 50만개 기준:
연간 \$569.89 약 65만원:
캡처당 약 1.3원 30

3-6. GRAZBIT - 공식 블로그

OFFICIAL GRABZIT BLOG

Read our blog to keep up to date with what we are working on and to discover new and interesting ways you can use GrabzIt in your own projects. You can also subscribe to our blogs [RSS feed](#) to get notified about our latest posts.



Lots of new features coming in version 3.2

Friday, June 23, 2017

As you may be aware we sent a survey out a few weeks ago, asking if you wanted the ability to automatically upload captures to Amazon S3 the responses came back overwhelmingly that you do!

So that is what we have done all of our client libraries will soon support automatically uploading captures to Amazon S3, Dropbox, FTP and WebDav. To do this you will need to create a export URL that can then be passed to our API.

But we decided to add a few more features while we are at it!

Web Scraper Browser Upgraded!

Friday, June 09, 2017

Hot on the heels of our last two releases; the new HTML and URL to DOCX service and an updated version of our capture software, which has updated browser base code and fixes for improving rendering quality.

Do you want to create captures faster?

Thursday, May 25, 2017

GrabzIt is excited to announce we have just released a geo-targeting system! This applies to all captures that convert URLs, such as converting URLs to images, PDF's and Word documents

DOCX conversions are coming!

Friday, May 19, 2017

You probably haven't heard much from GrabzIt for a while, well that's not because we haven't been busy...we have been beavering away for the last three months creating a

3. 정리

| | openhtmltopdf | DocRaptor | ITEXT | GRAZBIT |
|----|-------------------------------|--------------------|-----------------------|-------------------------------|
| 특징 | Open Source with readable API | Just functionality | Powerful control | Low price with easy interface |
| 장점 | 높은 확장가능성 및 상세한 레퍼런스 | 적당한 기능 및 포맷 컨트롤 | 광범위한 개발 문서 및 레퍼런스 | 저렴한 비용 |
| 단점 | 안정성 | - | Overkill & 무거움 | 부실한 개발 문서 |
| 비용 | 무료 | 건당 최대 23원 | 서버당 라이선스 비 용 386만원 | 건당 1.3원 |