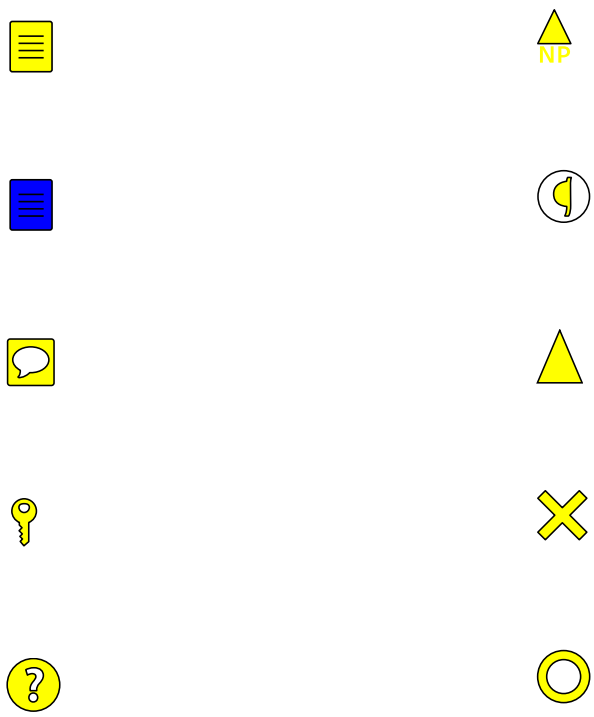
PSPDFKit Annotation Test Page

PSPDFKit supports almost all PDF annotation types as defined in the PDF Reference 1.7. Here you'll find examples of the supported types.

Note Annotations (PSPDFNoteAnnotation)

A note annotation is represented as an overlay icon attached to a point on the PDF. On touch, it shows the text set inside. PSPDFKit provides custom images for the types that are defined in the PDF Reference (Comment, Key, Note, Help, NewParagraph, Paragraph, Insert) and some other commonly used types. Note images are dynamically colored.

```
// Example how to create note annotations programmatically.
PSPDFNoteAnnotation *noteAnnotation = [PSPDFNoteAnnotation new];
noteAnnotation.boundingBox = CGRectMake(100, 100, 50, 50);
noteAnnotation.contents = @"This is a test for the note annotation.";
[document addAnnotations:@[noteAnnotation] forPage:0];
```



Highlight Annotations (PSPDFHighlightAnnotation)

Highlight annotations (“Text markup annotations” in the PDF Reference) work on text rects and should thus only be created if there’s actual text on the PDF and not an image. PSPDFKit currently supports Highlight, Underscore and Strikeout. Highlight annotations have both a boundingBox and an array of rects (the highlighted words). PSPDFKit has some helpers that will calculate the smallest fitting boundingBox for the array of rects.

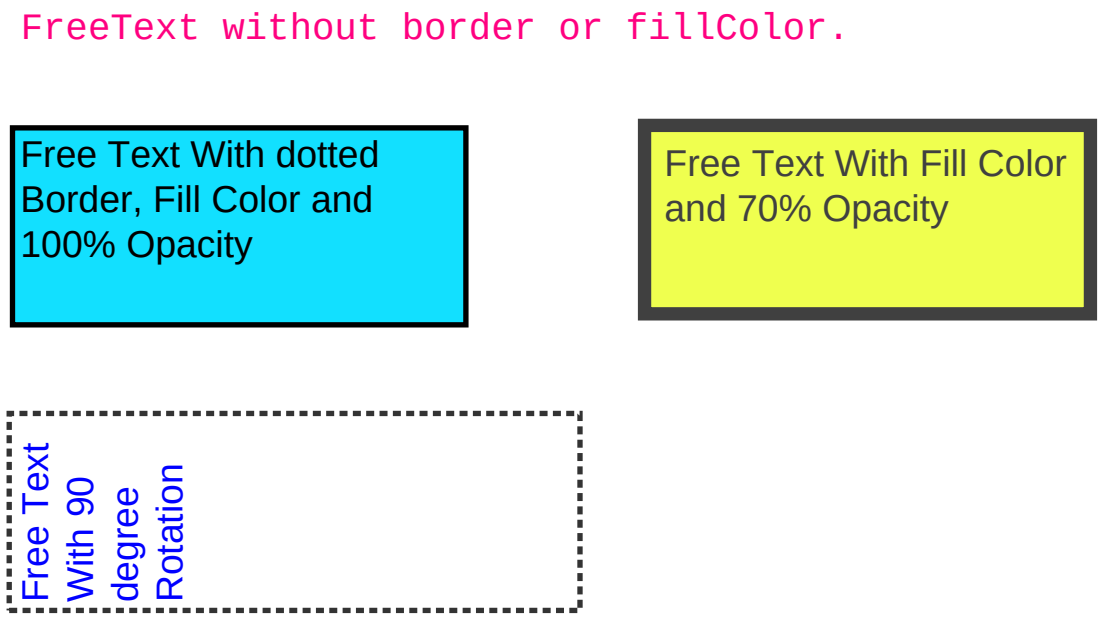
If a highlight annotation contains a note, PSPDFKit will display them slightly different to show this.

```
// Example how to dynamically highlight certain words from the page.
for (NSUInteger idx = 0; idx < 6; idx++) {
    PSPDFWord *word = [[document textParserForPage:0].words objectAtIndex:idx];
    PSPDFHighlightAnnotation *annotation = [[PSPDFHighlightAnnotation alloc]
initWithHighlightType:PSPDFHighlightAnnotationHighlight];
    CGRect boundingBox;
    annotation.rects = PSPDFRectsFromGlyphs(word.glyphs, [document
pageInfoForPage:0].pageRotationTransform, &boundingBox);
    annotation.boundingBox = boundingBox;
    [document addAnnotations:@[annotation] forPage:0];
}
```

FreeText Annotations (PSPDFFreeTextAnnotation)

FreeText annotations show their text directly on the page, they don't have a hidden Note field. FreeText annotations can have various options, border, fillColor, fontSize/Name, etc.

```
// Create a FreeText annotation
PSPDFFreeTextAnnotation *freeText = [PSPDFFreeTextAnnotation new];
freeText.contents = @"This is a test.\n1\n2\n3\n4\n5";
freeText.boundingBox = CGRectMake(100, 100, 400, 200);
freeText.fillColor = UIColor.yellowColor;
freeText.fontSize = 40;
[document addAnnotations:@[freeText] forPage:0];
```



Stamp Annotation (PSPDFStampAnnotation)

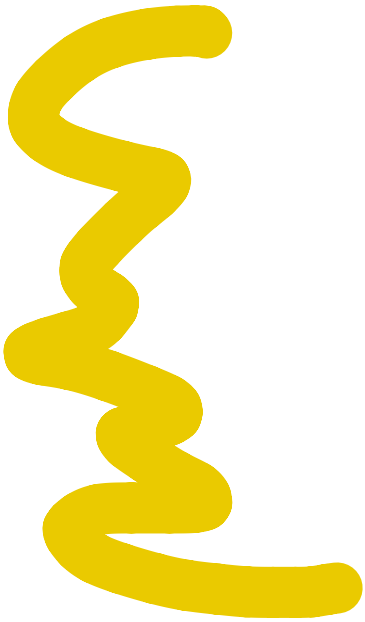
Stamp annotations can contain a whole PDF as dataSource. In PSPDFKit this can be read, and the framework supports both the old default annotations (ACCEPTED, REJECTED, ...), a mode that creates custom stamps, and (more common) image stamps. The Image feature of PSPDFKit simply uses stamp annotations as well.

```
// Add an image stamp annotation.
PSPDFStampAnnotation *stampAnnotation = [PSPDFStampAnnotation new];
stampAnnotation.image = [UIImage imageNamed:@"testimage.png"];
stampAnnotation.boundingBox = stampAnnotation.image.size;
stampAnnotation.subject = @"Image Stamp";
[document addAnnotations:@[stampAnnotation] forPage:0];
```

Ink Annotations (PSPDFInkAnnotation)

An ink annotation (PDF 1.3) represents a freehand “scribble” composed of one or more disjoint paths. PSPDFKit has several helper functions to work with those points/lines, see PSPDFInkAnnotation.h for details. A border set will be applied to the whole line.

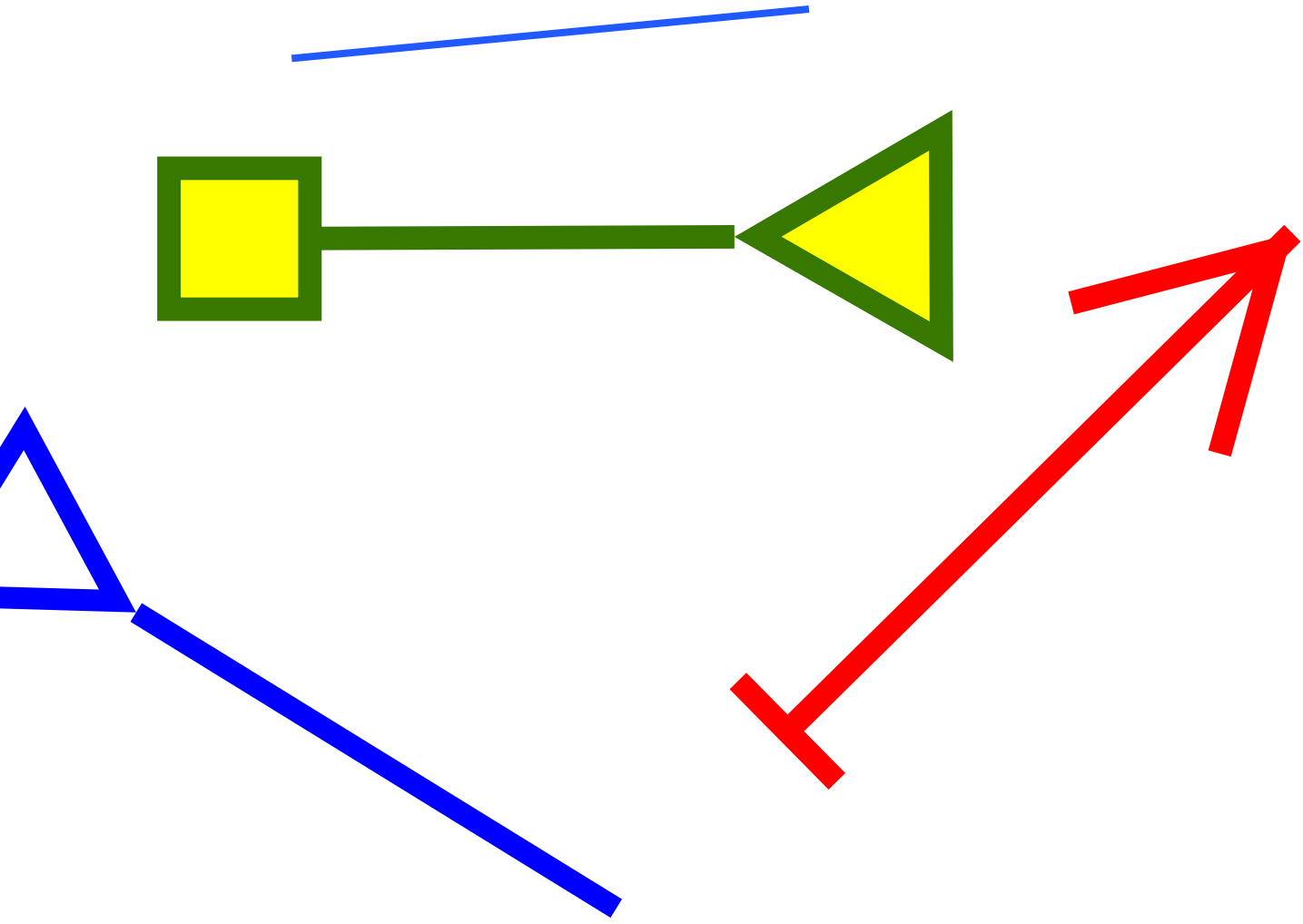
```
// Programmatically create an Ink annotation
PSPDFInkAnnotation *annotation = [PSPDFInkAnnotation new];
// Example how to create a line rect. Boxed is a macro for [NSValue valueWithCGRect:]
NSArray *lines = @[@[BOXED(CGPointMake(100,100)), BOXED(CGPointMake(100,200)),
BOXED(CGPointMake(150,300))], // first line
                   @[BOXED(CGPointMake(200,100)), BOXED(CGPointMake(200,200)),
BOXED(CGPointMake(250,300))]];  // second line
// Convert view line points into PDF line points.
PSPDFPageInfo *pageInfo = [document pageInfoForPage:targetPage];
CGRect viewRect = [UIScreen mainScreen].bounds; // this is your drawing view rect - we
don't have one yet, so lets just assume the whole screen for this example. You can also
directly write the points in PDF coordinate space, then you don't need to convert, but
usually your user draws and you need to convert the points afterwards.
annotation.lineWidth = 5;
annotation.lines = PSPDFConvertViewLinesToPDFLines(lines, pageInfo.pageRect,
pageInfo.pageRotation, viewRect);
annotation.color = [UIColor colorWithRed:0.667 green:0.279 blue:0.748 alpha:1.000];
[document addAnnotations:@[annotation] forPage:targetPage];
```



Line Annotations (PSPDFLineAnnotation)

The purpose of a line annotation (PDF 1.3) is to display a single straight line on the page. PSPDFKit supports all line endings as defined in the PDF Reference. (The most common type will be the arrow). Apple's Preview.app has only limited support for line annotations and will not understand all line ending types. Adobe Acrobat supports them.

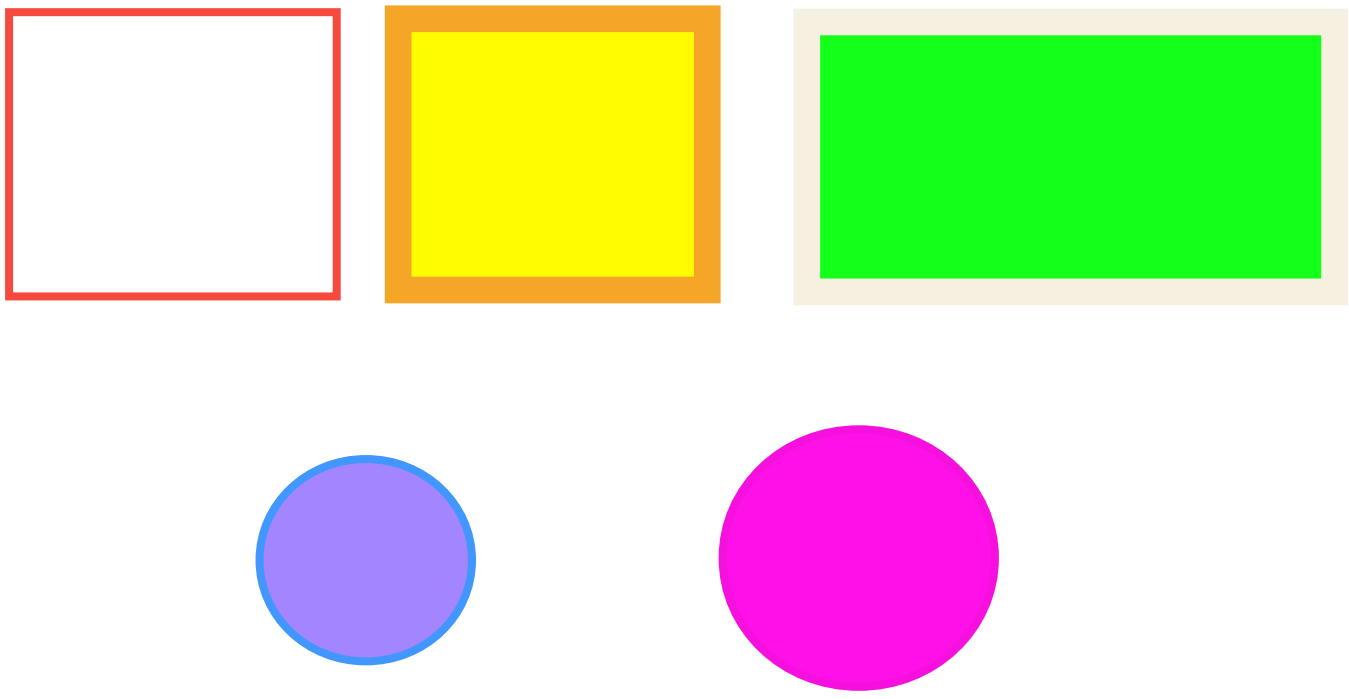
```
// Create a new line annotation
PSPDFLineAnnotation *lineAnnotation = [PSPDFLineAnnotation new];
lineAnnotation.fillColor = fillColor; // Fills the area inside the arrow.
lineAnnotation.color = color;
lineAnnotation.lineWidth = 5.f;
lineAnnotation.point1 = CGPointMake(0.f, 0.f);
lineAnnotation.point2 = CGPointMake(100.f, 100.f);
lineAnnotation.lineEnd2 = PSPDFLineEndTypeClosedArrow; // Add arrow at the end.
```



Rectangle/Oval Annotations (PSPDFShapeAnnotation)

Square and circle annotations (PDF 1.3) shall display, respectively, a rectangle or an ellipse on the page. Color and FillColor can be set independently, but the annotation shares a common opacity value.

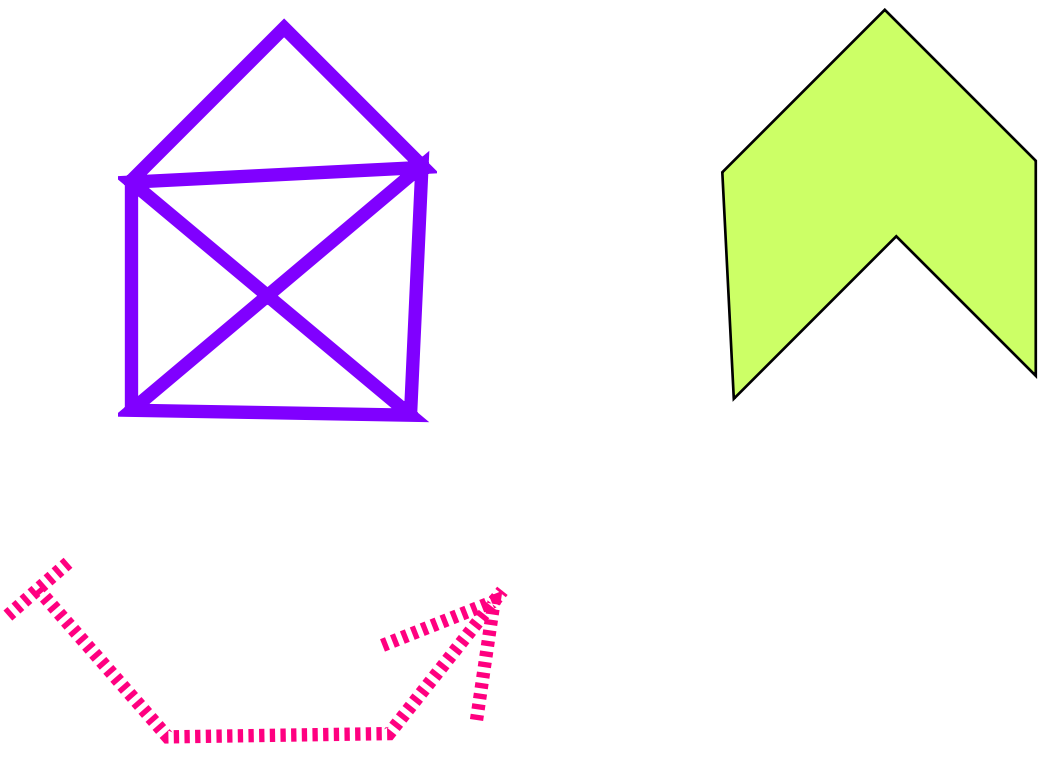
```
// Create a semi-transparent, filled Square shape annotation
PSPDFShapeAnnotation *annotation = [[PSPDFShapeAnnotation alloc]
initWithShapeType:PSPDFShapeAnnotationSquare];
annotation.boundingBox = CGRectInset([document
pageInfoForPage:targetPage].rotatedPageRect, 100.f, 100.f);
annotation.color = [UIColor colorWithRed:0.0f green:0.3f blue:0.f alpha:1.f];
annotation.fillColor = annotation.color;
annotation.alpha = 0.5f;
[document addAnnotations:@[annotation] forPage:targetPage];
```



Polyline/Polygon Annotations (PSPDFPolygonAnnotation)

Polygon annotations (PDF 1.5) display closed polygons on the page. Such polygons may have any number of vertices connected by straight lines. Polyline annotations (PDF 1.5) are similar to polygons, except that the first and last vertex are not implicitly connected.

```
// Create a PolyLine annotation
PSPDFPolygonAnnotation *polyline = [[PSPDFPolygonAnnotation alloc]
initWithPolygonType:PSPDFPolygonAnnotationPolyLine];
polyline.points = @[[NSValue valueWithCGPoint:CGPointMake(52, 633)], [NSValue
valueWithCGPoint:CGPointMake(67, 672)], [NSValue
valueWithCGPoint:CGPointMake(131, 685)], [NSValue
valueWithCGPoint:CGPointMake(178, 654)], [NSValue
valueWithCGPoint:CGPointMake(115, 622)]];
polyline.color = [UIColor colorWithRed:0.0 green:1.f blue:0.f alpha:1.f];
polyline.fillColor = UIColor.yellowColor;
polyline.lineEnd2 = PSPDFLineEndTypeClosedArrow;
polyline.lineWidth = 5.f;
[document addAnnotations:@[polyline] forPage:targetPage];
```



File Annotations (PSPDFFileAnnotation)

A file attachment annotation (PDF 1.3) contains a reference to a file, which typically shall be embedded in the PDF file (see 7.11.4, “Embedded File Streams”). PSPDFKit will offer QuickLook on those annotation types (and thus also Open In...)

```
File annotations are currently readonly.
```

Video Annotations (PSPDFRichMediaAnnotation, PSPDFScreenAnnotation)

A screen annotation (PDF 1.5) specifies a region of a page upon which media clips may be played. It also serves as an object from which actions can be triggered. PSPDFKit also supports the matching Rendition Actions to control the video play state.

RichMedia annotations are defined in the ISO32000 Adobe Supplement and are the modern way of embedding video content. PSPDFKit also supports the matching RichMediaExecute action to control video state.

```
Video Annotations are currently readonly.
```

Caret Annotations (PSPDFCaretAnnotation)

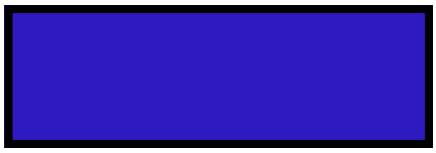
A caret annotation (PDF 1.5) is a visual symbol that indicates the presence of text edits, and share much similarity with note (text) annotations.

```
Caret annotations are currently readonly.
```

Widget Annotations (PSPDFWidgetAnnotation)

Widget annotations are mostly used for interactive forms or buttons. PSPDFKit doesn't yet support AcroForms, but supports the widget type to execute actions. This sometimes is used in combination with video/audio to provide play/pause buttons. The rendering of Widgets is covered in an appearance stream which basically is a second PDF.

```
Widget annotations are readonly.
```



Sound Annotations (PSPDFSoundAnnotation)

A sound annotation (PDF 1.2) shall analogous to a text annotation except that instead of a text note, it contains sound recorded from the computer's microphone or imported from a file. When the annotation is activated, the sound shall be played. The annotation shall behave like a text annotation in most ways, with a different icon.

```
Sound annotations are currently readonly.
```

Link Annotation (PSPDFLinkAnnotation)

A link annotation represents either a hypertext link to a destination elsewhere in the document or a different action. PSPDFKit supports all common action types, including Named Actions like Next, Previous, FirstPage, LastPage, Remote GoTo (to link to a different document) and even basic JavaScript (for page changes).

PSPDFKit also supports a variety of multimedia extension to the link annotation type with using ‘pspdfkit://’ as the URL handler. Visit the pspdfkit.com website to learn more.

```
// Create a customized link annotation that will show an image.
PSPDFLinkAnnotation *annotation = [[PSPDFLinkAnnotation alloc]
initWithLinkAnnotationType:PSPDFLinkAnnotationImage]; annotation.URL = [NSURL
URLWithString:[NSString stringWithFormat:@"pspdfkit://[contentMode=
%d]localhost/TokenTest/exampleimage.jpg", UIViewContentModeScaleAspectFill]];
// annotation frame is in PDF coordinate space.
annotation.boundingBox = [self.document pageInfoForPage:0].rotatedPageRect;
[document addAnnotations:@[annotation] forPage:targetPage];
```

First Page